uc3m | Universidad Carlos III de Madrid

Grado en Ingeniería de Sistemas Audiovisuales
(2017-2018)

Trabajo Fin de Grado

"Estudio de descriptores y algoritmos para la localización de objetos en imágenes"

Jorge Sánchez Alberto

Tutor: Miguel Ángel Fernández Torres

Leganés, 25 de septiembre de 2018

RESUMEN

Durante los últimos años el área de la Visión Artificial (*Computer Vision*) se ha convertido en un campo de vanguardia. El crecimiento del estudio y la experimentación en este sector se deben a que cada vez se pueden dar más usos al análisis y procesamiento de la imagen.

El proyecto que se presenta a continuación consiste en el estudio y desarrollo de un sistema para la localización de objetos en imágenes. Este sistema muestreará un conjunto de ventanas sobre cada una de las imágenes, teniendo en cuenta la información a priori aportada por un histograma de localizaciones y relaciones de aspecto de los objetos. A continuación, extraerá características sobre las ventanas que permitan su posterior clasificación, dependiendo de si contienen un objeto o pertenecen al fondo de la escena.

Sobre la base de un conjunto de descriptores, se plantean tres alternativas de diseño del sistema, basadas en clasificadores de tipo Máquina de Vectores Soporte (SVM), los cuales han presentado habitualmente un rendimiento óptimo en el estado del arte. A partir del análisis de las tres alternativas propuestas, se concluye que la arquitectura que toma como entrada descriptores más tradicionales, basados en contrastes de color e información acerca de los bordes de la imagen, es la que ofrece una mejor localización de los objetos, siendo crítico el buen funcionamiento de la etapa de muestreo de ventanas candidatas a objeto sobre las imágenes para lograr este objetivo.

Palabras Clave: localización de objetos, histograma de localizaciones y relaciones de aspecto, extracción de características, SVMs

DEDICATORIA

A toda mi familia. En especial a mi madre sin su cariño y su confianza en mi nada de esto hubiera sido posible. Y a mi abuela porque este sueño es de los dos.

Y mi eterno agradecimiento a mi tutor Miguel Ángel, porque no todo el mundo haría lo que tú has hecho por mí y te aseguro que eso el resto de mi vida lo tendré en cuenta.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. MOTIVACIÓN Y OBJETIVOS DEL PROYECTO

1.1 Motivación

Durante los últimos años el área de la Visión Artificial (*Computer Vision*) se ha convertido en un campo de vanguardia. El crecimiento del estudio y la experimentación en este sector se deben a que cada vez se pueden dar más usos al análisis y procesamiento de la imagen. Además, gracias al avance y aumento de la tecnología, estas técnicas, computacionalmente complejas o que antes podían suponer un coste elevado, pueden ahora ser aplicadas a otros sectores como el automovilístico, el de la telefonía móvil y el de la seguridad en el hogar.

Se puede definir la Visión Artificial como una rama de la Inteligencia Artificial que basa sus estudios en el desarrollo de técnicas para el procesamiento y análisis de las características extraídas a partir de las imágenes digitales [1].

Las principales aplicaciones de la Visión Artificial son:

- Identificación y análisis de objetos.
- Localización de los objetos en el espacio.
- Establecimiento de relaciones entre objetos en el espacio.
- Modelado y reconstrucción de objetos en tres dimensiones.
- Detección y representación de partes específicas de un objeto (esquinas, piezas concretas, etc.)

El proyecto que se presenta a continuación consiste en el estudio y desarrollo de un sistema para la localización de objetos en imágenes. Este sistema muestreará un conjunto de ventanas sobre cada una de las imágenes, extrayendo características que permitan su posterior clasificación, dependiendo de si contienen un objeto o pertenecen al fondo de la escena.

Un sistema de localización de objetos puede resultar de gran utilidad en una gran variedad de escenarios. Efectivamente, nos permite detectar desde objetos de grandes dimensiones, como monumentos, hasta elementos como el logotipo de una marca de moda. Un ejemplo de aplicación real, hoy en día presente en los dispositivos móviles más recientes, es el uso de una o varias cámaras digitales para el modelado y reconocimiento de la cara de los usuarios. Otro caso más simple es el reconocimiento del código de barras de un producto, detectando primero la ventana o *bounding box* (BB) que contiene el mismo para su correcto procesado posterior.

Otros usos populares de localizadores de objetos incluyen la tecnología de 'ojo de halcón' en deportes como el tenis. Ésta consiste en localizar la pelota de tenis cuando bota en la pista, de forma que se pueda saber si el área completa de la pelota está fuera de los límites del campo o, por el contrario, no ha salido por completo. Este sistema se basa en la toma en directo de varias imágenes y la posterior triangulación de la posición final del objeto. Dado su gran nivel de precisión, se ha comenzado a evaluar su uso en otros deportes [2].

Uno de los sectores en los que más se está investigando con imágenes actualmente es el de la seguridad. Un ejemplo de aplicación concreto son los 'sistemas de detección de entorno', que cada vez incorporan más coches para prevenir atropellos de peatones o ciclistas. Para un sistema de este tipo es necesario, como mínimo, el empleo de dos cámaras de alta resolución colocadas en la parte alta del coche, a la altura del

limpiaparabrisas [3]. Éste sería solamente uno de sus múltiples usos: el gran objetivo que se persigue en el sector automovilístico es el de la conducción automática.

A continuación, se pasará a definir el objetivo principal del proyecto, así como los objetivos específicos que se han de cumplir para alcanzarlo.

1.2 Objetivos

El objetivo fundamental del proyecto es la detección y localización de objetos en imágenes. Para ello, se estudian diferentes descriptores y algoritmos, con el propósito de proponer un sistema basado en aquellas técnicas que ofrezcan mejores prestaciones.

Este objetivo fundamental puede describirse atendiendo a los siguientes objetivos específicos. Los tres primeros se corresponden con las diferentes etapas del sistema:

1. Muestreo de ventanas candidatas a objeto en la imagen: Esta etapa es crucial en el sistema, pues es el primer paso a la localización de los objetos. En este proyecto, se estudiará el uso de una técnica de muestreo de ventanas basada fundamentalmente en un histograma de localizaciones y relaciones de aspecto.

2. Extracción de características: En esta etapa se estudiarán e implementarán diferentes descriptores para la representación de las ventanas muestreadas en el primer paso para cada imagen, candidatas a contener un objeto. Estas características serán genéricas, es decir, independientes de la categoría semántica de los objetos a localizar, para que el sistema sea capaz de detectar cualquier tipo de objeto.

3. Localización de los objetos: En esta etapa, se hará uso de un método de clasificación conocido y usado frecuentemente en el estado del arte, las Máquinas de Vectores Soporte (SVMs), para la clasificación de las ventanas candidatas, representadas mediante características, en ventanas que contienen un objeto y ventanas que pertenecen al fondo de la imagen.

Los objetivos específicos enumerados hasta ahora se representan a continuación, en el diagrama de la Figura 1.

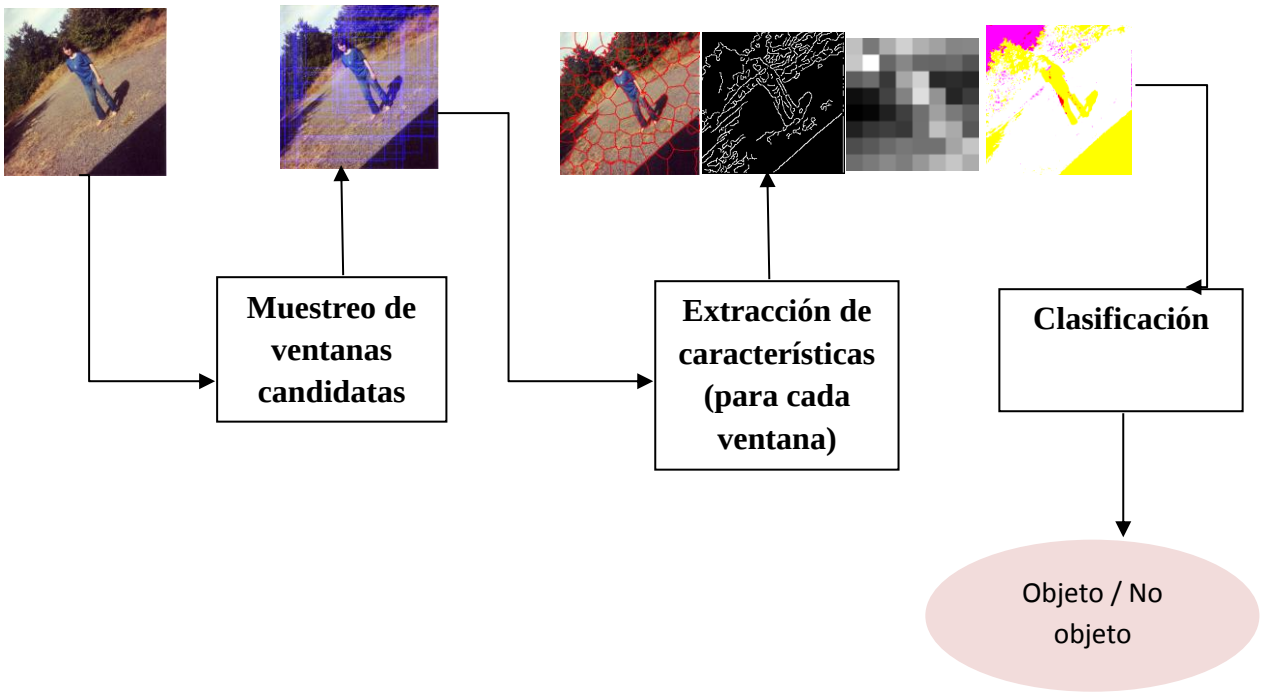


Figura 1. Diagrama básico de un sistema de localización de objetos.

Como objetivo específico final del proyecto, se tienen en cuenta la evaluación y análisis del sistema implementado:

4. Evaluación: Se procederá a determinar de manera objetiva la eficiencia del sistema desarrollado en el proyecto. Con la intención de poder aplicar estos conceptos a cualquier imagen, se aplican sobre una muestra representativa formada por la base de datos PASCAL VOC [21].

1.3 Estructura de la memoria

La memoria del proyecto se divide en 7 capítulos. A continuación, se ofrece una descripción breve de cada uno de ellos:

- Capítulo 1: Motivación y objetivos. En primer lugar, se introducen las razones por las que se realiza el proyecto. A continuación, se define el objetivo fundamental que se desea alcanzar y los objetivos específicos para el planteamiento del proyecto.

- Capítulo 2: Planteamiento del problema. Se introducen las técnicas relacionadas con la detección de objetos en el estado del arte, describiendo los fundamentos teóricos de aquellas que se utilizarán para implementar el sistema estudio de este trabajo.

- Capítulo 3: Diseño de la solución técnica. Se detallan las herramientas utilizadas para el desarrollo del proyecto, y se explican de cada uno de los bloques que forman el sistema.

- Capítulo 4: Resultados y evaluación. Uno de los capítulos más importantes de la memoria, porque es donde se contrastan las hipótesis formuladas en el planteamiento y tenidas en cuenta en el diseño de la solución técnica con los resultados obtenidos. Al final se llevará a cabo un análisis de errores, con vistas a introducir mejoras en el sistema implementado.

- Capítulo 5: Planificación del trabajo y presupuesto. Se enumeran las tareas que se han llevado a cabo en el trabajo y los plazos que se han establecido para las mismas. Además, se incluye un presupuesto para la realización completa del proyecto.

- Capítulo 6: Conclusiones y líneas futuras. En esta sección se resumen las conclusiones extraídas del proyecto y se detallan las posibles mejoras o líneas futuras que podrían tenerse en cuenta para la continuación del trabajo.

- Capítulo 7: Bibliografía. Se citan las fuentes y artículos que se han usado a lo largo del proyecto como referencia.

2. PLANTEAMIENTO DEL PROBLEMA

2.1 Introducción

El problema de la detección y localización de objetos ha sido tratado ya durante más de dos décadas el ámbito de la Visión Artificial [7,8]. A pesar de existir una gran variedad de propuestas para solucionar este problema, todas ellas tienen algunos puntos en común.

Cuando se trata de la localización de un objeto del que se conoce previamente su clase (ej. coche), los sistemas suelen estar basados en las características del propio elemento (color, forma [4], tamaño...). Conocidas las características del objeto, no es necesario saber su localización espacial a priori.

Para explicar con un sencillo ejemplo en qué consisten las características de un objeto se eligen primero unas características (color, forma, tamaño). A continuación, procedemos a describir de un par de ejemplos de objetos:

- Balón de baloncesto: naranja, redondo, 20-25 cm de diámetro.
- Televisión: negra, rectangular y 32 pulgadas.

Uno de los algoritmos que más se han utilizado en la detección de objetos y en concreto para la detección de caras es el de Viola & Jones, [5]. Este detector utiliza un clasificador en cascada, utilizando diferentes filtros que permiten determinar si la distribución de los elementos presentes en la imagen se corresponde con la de los rasgos característicos del objeto a detectar (en el caso de la cara: ojos, nariz, boca, etc.)

Sin embargo, cuando se trata de detectar y analizar cualquier objeto genérico, independientemente de su categoría, han de tenerse en cuenta propiedades comunes a todos ellos, que permitan separarlos del fondo de la imagen.

Atendiendo a los primeros ejemplos propuestos, en este proyecto se quiere localizar tanto un balón como una televisión, por lo que la búsqueda de estos no se puede basar en las características de los objetos. Se han de buscar características en común: ambos tienen una forma definida y un contorno cerrado; además, en general, tienden a tener un color distinto al color del fondo.

El primer paso a la localización de objetos en una imagen consiste en el análisis de ventanas candidatas. Una ventana candidata selecciona un área de la imagen de la cual se estudiarán sus píxeles. Habitualmente, para localizar un objeto, se precisa evaluar un número de ventanas alto. Por ello, es importante seleccionar una técnica de muestreo robusta y computacionalmente eficiente.

Para determinar si las ventanas muestreadas contienen un objeto o pertenecen al fondo de la imagen, es necesario representarlas mediante características, para su posterior clasificación. En esto consiste el objetivo principal de nuestro sistema, en clasificar zonas de la imagen en función de si contienen o no un objeto.

Una vez localizados los objetos, el siguiente paso podría ser reconocerlos, es decir, asignarlos a una categoría semántica. Esta tarea, sin embargo, no forma parte de los objetivos específicos de este proyecto.

2.2 ¿Qué es un objeto? ¿Dónde se sitúa un objeto?

De acuerdo con uno de los artículos más populares sobre localización de objetos en imágenes [6], un objeto se puede definir de la siguiente manera:

"Los objetos son elementos que se caracterizan por ser independientes, con un contorno de límites continuos y un centro bien definido, tales como vacas, automóviles y teléfonos. Como caso contrario se encuentran los elementos o zonas del fondo de la imagen del paisaje, con un patrón de textura característico, pero sin un contorno compacto, tales como cielo, hierba y camino."

La mayoría de los estudios sobre la detección de objetos establecen que todo objeto que se desee localizar debe cumplir al menos con una de estas tres características distintivas:

1. Un contorno cerrado apreciable respecto a las características del fondo [7,8].

2. Una apariencia distinta a la de su vecindad o entorno más próximo [6].

3. Constituir una zona llamativa o saliente de la escena [6,7,8].

A la hora de localizar un objeto en una imagen, se ha de escoger un procedimiento de muestreo de ventanas candidatas a contener el mismo. Existen diferentes enfoques para resolver este problema, de los cuáles uno de los más conocidos es el método de la ventana deslizante [11]. Este método consiste en posicionar una ventana de un tamaño determinado en la esquina superior izquierda de la imagen, e ir desplazándola de izquierda a derecha y de arriba a abajo de forma que recorra todas las zonas de ésta. Para poder capturar objetos a diferente escala, es necesario repetir este proceso, utilizando tamaños de ventana diferentes, lo que hace que la ventana deslizante sea un método muy poco eficiente.

El problema a resolver para localizar el objeto será el de determinar la ventana o Bounding *Box* (BB) más acotada posible, dentro de la cual se encontrarán todos los puntos de interés del objeto, tal y como se muestra en los ejemplos de la Figura 2.

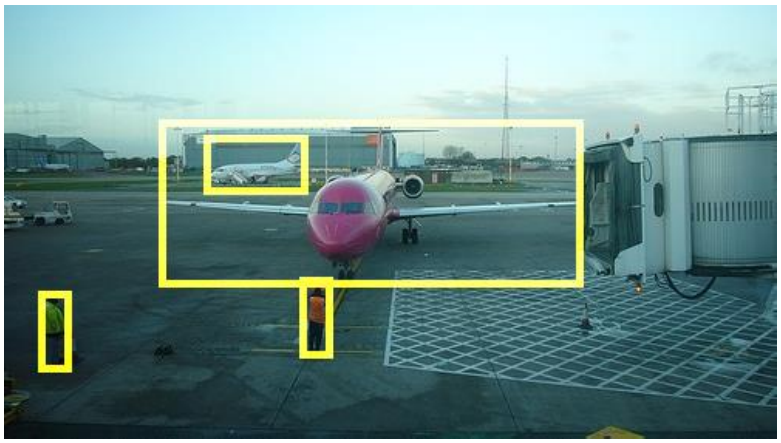


Figura 2. Ejemplo de fotografía de un avión en la base de datos PASCAL VOC [21], incluyendo ventanas o bounding boxes (BB) que indican los objetos a localizar.

En este proyecto utilizaremos un método de muestreo de ventanas diferente al de ventana deslizante, basado en un histograma de localizaciones y relaciones de aspecto de los

objetos. Aunque la zona de localización de un objeto es casi siempre variable, se observan algunas tendencias a la hora de situar un objeto cuando se toma una fotografía. En el caso más general, sin importar quién realiza la fotografía o qué se pretende captar, se puede deducir que el objeto que se quiere capturar estará en posiciones centradas para seleccionar su área por completo.

Una de las formas más frecuentes de ordenar los objetos en la imagen es mediante la "regla de los tercios" [9,10]: "La regla de los tercios es un medio simple de aproximación a la proporción áurea (...) y que trata la distribución del espacio dentro de la imagen que genera una mayor atracción respecto al centro de interés." A partir de esto establecemos nueve sectores de interés para situar el área del objeto, tal y como se muestra en la figura siguiente:

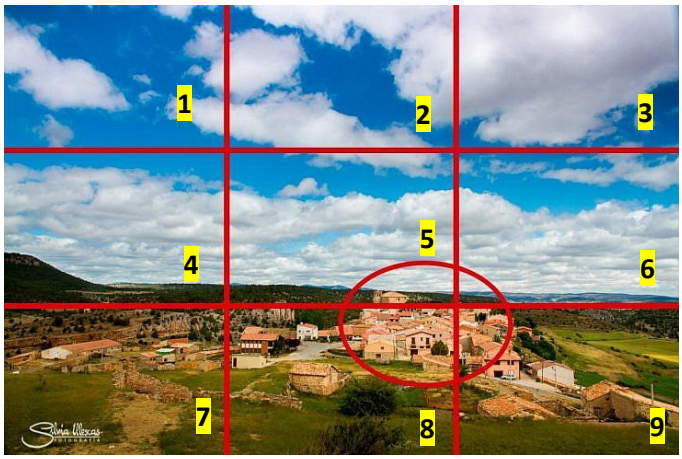


Figura 3. Regla de los tercios [20]

El orden de localización es tal como sigue: el primer sector se corresponde con la esquina superior izquierda y, recorriendo de izquierda a derecha la imagen, y de arriba abajo, se van enumerando el resto de los sectores. En el caso de la Figura 3, el objeto de la imagen seria el pueblo, que está localizado en los sectores 5, 6, 8 y 9.

Basándonos en la regla de los tercios, el procedimiento a seguir en nuestro sistema será muestrear ventanas principalmente en aquellas zonas con mayor probabilidad de objeto, de acuerdo a un conjunto de imágenes del cual se conoce previamente la localización de los objetos, anotada por medio de ventanas o *BBs* a las que en numerosas ocasiones nos referiremos como muestras positivas. Muestreando en zonas de la imagen más probables de contener un objeto intentaremos reducir el número de ventanas a muestrear y, con ello, el coste computacional y el tiempo de ejecución del método.

Para determinar por medio de probabilidades de ocurrencia en qué sectores es más frecuente encontrar un objeto se utilizará como herramienta el histograma. El histograma es una representación gráfica de datos, valores y otros elementos numéricos de interés, que tiene como objetivo la visualización, interpretación y uso de resultados.

El histograma que vamos a utilizar contendrá las probabilidades e ocurrencia de objetos en cada uno de los sectores de la regla de los tercios, subdivididas a su vez en probabilidades basada en las relaciones de aspecto asociadas a los mismos.

2.3 Extracción de características

Cuando se desconoce si la ventana candidata a evaluar contiene un objeto o, en cambio, pertenece al fondo, se recurre a la definición de objeto introducida en el apartado para escoger y diseñar un conjunto de descriptores. Los descriptores son representaciones que se conciben con el propósito de cuantificar las características del área de las ventanas muestreadas en la imagen. Los valores que se obtengan para los descriptores deben de ser lo más representativos posibles de las clases positiva (objeto) y negativa (fondo), para una mejor separación de las mismas en la etapa de clasificación.

Para la representación de ventanas candidatas a objetos en este proyecto, se han extraído un conjunto de características que se apoyan en las técnicas y algoritmos de procesado de imagen que se describen a continuación, así como en el uso de diferentes espacios de color.

2.3.1 Gradiente de una imagen

Se ha observado que los objetos, que se diferencian del entorno por su color o contorno definido, destacan si se calcula la norma del gradiente de la imagen.

El gradiente es un vector donde sus elementos miden la rapidez en que los valores de los pixeles modifican con respecto de la distancia y en las direcciones x e y. Esto resulta de utilidad para detectar objetos, puesto que en la zona de la imagen donde hay un cambio de objeto a fondo, el gradiente será de un valor elevado.

Las derivadas de las direcciones, dx y dy, corresponderán con el número de pixeles entre dos puntos, siguiendo las ecuaciones:

$$\frac{\partial f(x,y)}{\partial x} = \frac{f(x+ d_x,y)- f(x,y)}{dx} \qquad (2.4)$$

$$\frac{\partial f(x,y)}{\partial y} = \frac{f(x+ d_y,y)- f(x,y)}{dy} \qquad (2.5)$$

Para detectar la presencia de una discontinuidad en el gradiente, se debe calcular el cambio en el gradiente en el punto, por ejemplo (km).

$$\Delta x = f(k+1, m) - f(k, m) \qquad (2.6)$$

$$\Delta y = f(k, m+1) - f(k, m) \qquad (2.7)$$

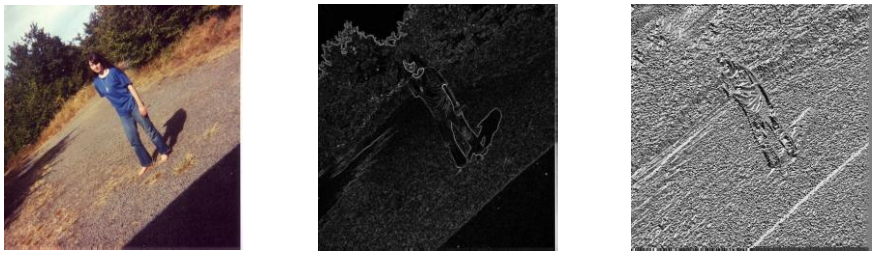


Figura 4. Ejemplo del cálculo de gradiente en una imagen. (a) Imagen original (b)Imagen de la dirección del gradiente (c) Imagen de la magnitud del gradiente en escala de grises.

Si se usa como ejemplo la Figura 4, se puede comprobar que el gradiente presenta los bordes del objeto y de aquellos elementos que sufran una variación en el valor de sus pixeles.

2.3.2 Extracción de bordes: Detector de Canny

Tal y como se ha introducido anteriormente, los objetos se caracterizan por bordes o contornos cerrados. Su detección en la imagen, por tanto, es esencial para la localización de los objetos. Aunque en el fondo de la imagen también encontramos bordes, estos son más cortos y de menor contraste, correspondientes a texturas. Estableciendo un umbral adecuado, estos bordes menos marcados se pueden eliminar, pudiendo definir una característica representativa de objeto basada en bordes, como la que se definirá en el siguiente capítulo.

Para la detección de bordes se ha utilizado el popular algoritmo de Canny [24]. Podemos distinguir diferentes fases en el detector de Canny. El primer proceso consiste en aplicar un filtro gaussiano para la eliminación de cualquier posible ruido en la imagen. A continuación, se debe encontrar el gradiente de intensidad de la imagen. Para ello se filtrará con dos operadores de Sobel, uno con dirección vertical, G_x, y otro con dirección horizontal, G_y. La magnitud y dirección del gradiente vienen dadas entonces por:

$$|G| = \sqrt{(G_x^2 + G_y^2)} \qquad (2.2)$$

$$\phi_G = arctan\left(\frac{G_y}{G_x}\right) \qquad (2.3)$$

A continuación, se determina si el valor de la magnitud de gradiente es más pequeño que al menos uno de sus vecinos, en la dirección calculada con la fórmula 2.3. Si cumple con este criterio se le asignará un valor de 0 a ese píxel, en caso contrario se le da el valor correspondiente a la magnitud del gradiente.

La imagen resultante del proceso anterior suele contener mucho ruido por lo que, para su eliminación, se recurre a la histéresis del umbral. El proceso toma como base la imagen generada en el paso anterior, se calculan la orientación de los bordes y se establecen dos umbrales de valores distintos, primario y nivel bajo. Finalmente, se filtra, eliminando las aristas que no alcancen el umbral alto, el primario.

Como se aprecia en la Figura 7(a), tras aplicar el proceso, el resultado es una imagen donde lo único que observa son los bordes de los objetos mejor definidos.

2.3.3 Espacios de color

El color es una característica importante a tener en cuenta siempre. Como se ha visto, sirve para distinguir entre dos objetos de clases diferentes. Para distinguir objeto de fondo, se puede calcular el contraste entre sus colores en un espacio de color determinado, tal y como se verá en la sección 3.4.

Un espacio de color es un sistema para dotar de manera objetiva de un valor al color. El modelo más común es el espacio de color RGB, que se distingue en que describe el valor del color mediante la mezcla de la cantidad de rojo, verde y azul que harían falta para formarlo.

Otro espacio de color comúnmente usado para la extracción de características es el espacio Lab. En este espacio, los colores se pueden describir en términos de matiz (color), luminosidad (brillo) y saturación (intensidad). El espacio de color Lab se basa en:

L = luminosidad Figura 5(b)

a = diferencia entre rojo y verde Figura 5(c)

b = diferencia entre amarillo y azul Figura 5(d)

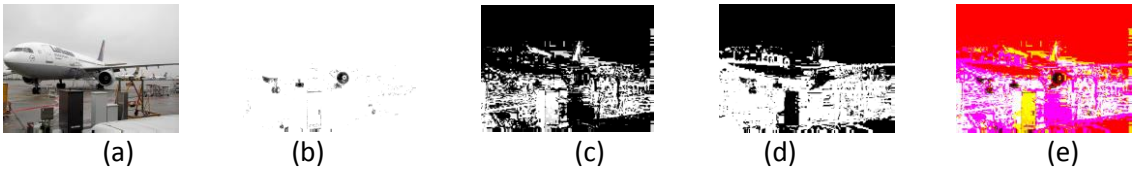


Figura 5. Ejemplo de una imagen en el espacio (a) Componente L (b) Componente a (c) Componente b (d) Componentes L, a y b superpuestas

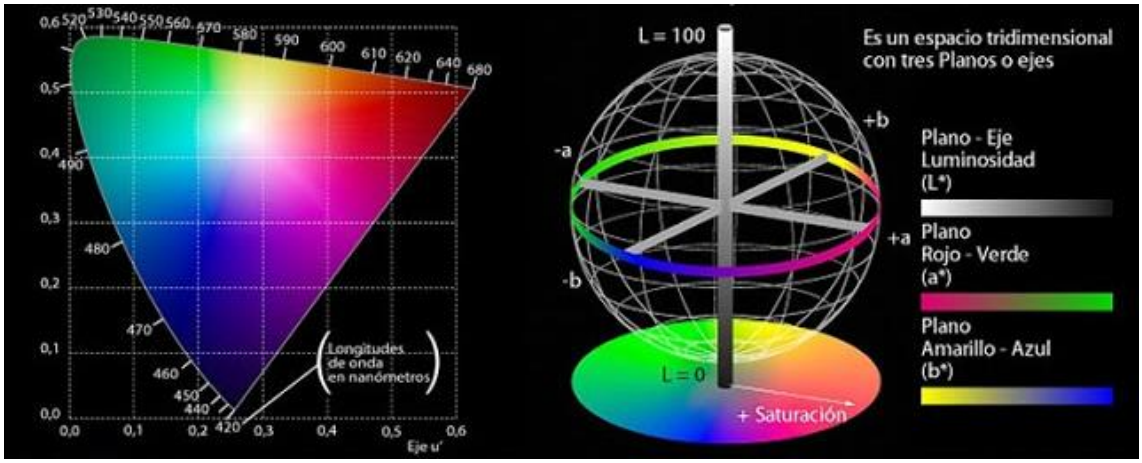


Figura 6. Espacio de color Lab [27]

En la Figura 6, se puede observar todo lo explicado en sobre el espacio de color Lab. Tal y como se muestra, su composición la forman 3 planos: luminosidad, distancio rojo-verde y distancia amarillo-azul.

2.3.4 Superpíxeles

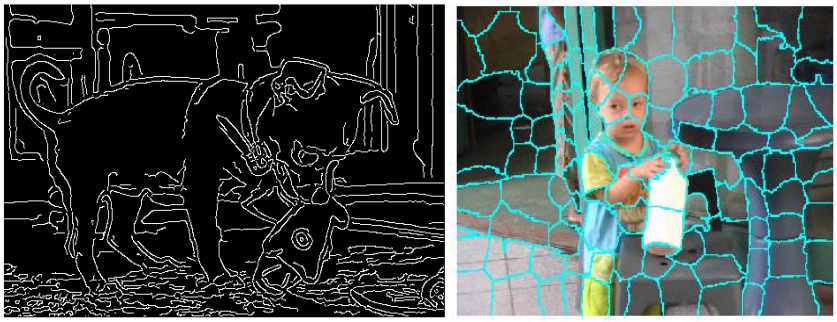
Dada una imagen, un superpíxel es la agrupación de varios píxeles de valores similares, pequeñas regiones de similar color o textura.

Para la creación de superpíxeles se emplea un algoritmo de iteración lineal simple (SLIC) [24], que permite al usuario especificar el número de superpíxeles en los que quiere dividir la imagen. Este número es una aproximación, el resultado será un valor similar, pero dependerá siempre de la imagen y sus rasgos a la hora de dividirla.

La técnica en la que se basa el algoritmo para determinar qué conjunto de píxeles forman un superpíxel consiste en agrupar regiones con valores similares dentro de la imagen, usando características como el contorno, color y la textura. De esta forma si la imagen presenta un objeto con un color distinto del fondo, no se dará el caso que un superpíxel esté formado por píxeles del fondo y del objeto ya que no tienen valores similares.

Por tanto, a ventaja o pista que ofrece esta técnica a la hora de localizar un objeto es que un superpíxel preserva los límites del mismo. Efectivamente, todos los píxeles que contiene un superpíxel pertenecerán casi siempre a un mismo objeto, por lo que un objeto se puede considerar como un conjunto de superpíxeles.

Como se observa en la Figura 7(b), la botella que sujeta el niño está compuesta por 2 o 3 superpíxeles. Además, los superpíxeles que conforman el contorno de la fuente están formados en su gran mayoría por píxeles pertenecientes al objeto.



(a) (b)

Figura 7. Fotografía (a) tras aplicar el detector de Canny a una imagen de ejemplo e imagen (b) dividida en superpíxeles

2.3 Etapa de clasificación: Máquinas de vectores soporte (SVM)

El clasificador es el bloque que encarga de asignar a qué clase pertenece la ventana candidata y decidir si contiene una instancia del objeto o no.

En esta etapa del sistema de localización de objetos, se asigna un score o valor probabilístico a las ventanas muestreadas sobre una imagen, candidatas a contener un objeto, determinando la confianza sobre la presencia o ausencia del mismo.

Los métodos que se pueden utilizar en esta etapa se encuentran dentro del campo del Aprendizaje Máquina o Machine Learning. El proceso que habitualmente siguen estos métodos en la fase de entrenamiento o aprendizaje es el siguiente:

1. Se define un modelo mediante un conjunto de parámetros o pesos y se establecen unos valores iniciales a estos.

2. El modelo recibe la información o características una o varias muestras nuevas a aprender.

3. Se modifican o ajustan los valores de los pesos, de acuerdo con un mecanismo de optimización.

4. Se repiten los pasos 2 y 3 para ajustar más la máquina a la tarea a resolver.

Los algoritmos utilizados para la clasificación pueden ser supervisados o no supervisados. La diferencia entre ambos tipos reside en el conocimiento previo de las etiquetas reales.

La clasificación supervisada es aquella en la que se conocen las etiquetas del conjunto de entrenamiento. El algoritmo trabaja con las variables de entrada asignando unas etiquetas a la salida, que en el caso óptimo serían iguales a las que se conocen, los verdaderos valores de las etiquetas. Todo esto se lleva a cabo con un conjunto de datos con los que el sistema "aprende" a asignar etiquetas a los datos de salida.

Los clasificadores más utilizados son los detectores o clasificadores binarios, que diferencian entre dos clases. En el campo de la localización de objetos la primera clase se corresponde con objeto (clase positiva) y la otra con no-objeto o fondo (clase negativa). Un ejemplo sencillo es una imagen donde hay un avión con cielo de fondo, cuando la ventana se posicione sobre un elemento especifico como el avión, el clasificador lo catalogara como objeto. Cuando la ventana contenga en su mayoría fondo se debe catalogar en la otra clase.

Algunos ejemplos de algoritmos de aprendizaje supervisado son:

1. Regresión Logística [25]
2. Máquinas de Vectores Soporte (SVMs) [13]
3. Regresión por mínimos cuadrados [26]

El aprendizaje no supervisado se da cuando no se cuentan con las etiquetas para el entrenamiento, como antes sí sucedía en el supervisado. Sólo se cuenta ahora con los datos de entrada, ya que tampoco se dispone de los de salida. Uno de los métodos más usados es el c*lustering*, en el que se estudia el conjunto completo de datos y se intentan dividir en grupos con características similares. Algunos algoritmos de aprendizaje no supervisado son:

1. Algoritmo de clustering
2. Descomposición en valores singulares
3. Análisis de componentes principales

Dentro del grupo de algoritmos de aprendizaje supervisado, se ha decidido utilizar como clasificador para el sistema de localización de objetos de este proyecto el de Máquinas de Vectores Soporte (SVM), un método tradicionalmente muy empleado y que ha permitido alcanzar buenas prestaciones tanto en sistemas como el que se presenta [13] como en otros problemas de clasificación de imágenes (reconocimiento de monumentos, detección de melanomas, reconocimiento de caras, etc.).

El algoritmo SVM determina el mejor hiperplano posible para separar todas las muestras de una clase respecto a las de la otra clase. El mejor hiperplano posible es el que ofrece un mayor margen de separación entre ambas clases.

La Figura 8 muestra, dado un problema de clasificación el hiperplano de separación entre los datos con valor positivo y los datos de valor negativo. Aplicándolo a este proyecto, tal y como se ha ido mencionando a lo largo de este capítulo, la clase positiva identificará

las ventanas que sí contienen un objeto y la clase negativa las ventanas que pertenecen al fondo.

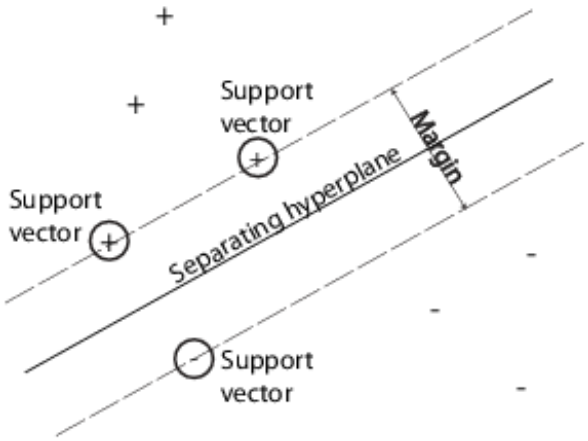


Figura 8. Diagrama de representación del hiperplano definido por el algoritmo SVM

2.5 Marco legislador

En el marco legislador se debe tener en cuenta la *Ley Orgánica 1/1982, de 5 de mayo, sobre protección civil del derecho al honor, a la intimidad personal y familiar y a la propia imagen* [A]; "Las imágenes usadas para las distintas aplicaciones no pueden ser usadas con otro objetivo o fuera de ese entorno, ya que, si no se posee la autorización de la persona con otros fines, no se deben almacenar esas imágenes si no es con el consentimiento de los propietarios de las fotos".

Por un lado, se ha de tener en cuenta la normativa relativa al uso de bases de datos de imágenes en aplicaciones o trabajos de investigación.

Para la realización del trabajo se ha utilizado una base de datos ya creada. Se ha utilizado la base de datos PASCAL VOC [21], que proporciona conjuntos de datos de imágenes para el reconocimiento de clases. Además, proporciona datos con anotaciones sobre las imágenes ya que, aunque no se ha realizado una clasificación por tipos de objetos si se podría haber hecho gracias a estas etiquetas. Su política de registro se basa en darse de alta por medio de un correo institucional, académicos inclusive, para evitar que una misma persona se registre varias veces. Las imágenes que componen PASCAL VOC se tomaron en gran medida de bases de datos públicas. También los propios autores aportan fotos. Para la realización del trabajo se ha seleccionado la versión de 2012, que consta de más de 11.570 imágenes de 20 clases distintas (perros, barcos, casas, aviones...).

Las condiciones de uso de PASCAL VOC, respecto a los derechos de la base de datos, son que la base de datos incluye algunas imágenes obtenidas de una web externa. El uso de esas imágenes debe respetar los términos de uso correspondientes de ese portal. Por lo tanto, si se quieren usar las imágenes para un uso externo, como una aplicación móvil, sí se debe pedir permisos, no a PASCAL VOC, sino a otra web que ayudó en la creación de la base de datos.

Figura 9. Muestras de las clases que ofrece PASCAL VOC [21]

La principal herramienta usada para el desarrollo del sistema ha sido MATLAB [14] que, como ellos mismos definen, es "una herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio". La plataforma de desarrollo ha sido Windows y la versión utilizada la 2018. MATLAB no es un software gratuito. Para su uso ha sido necesario darse de alta utilizando el convenio para estudiantes de la Universidad Carlos III de Madrid, que permite que un alumno de dicha universidad pueda descargarse el programa de forma gratuita si usa su cuenta de estudiante. Se han empleado funciones ya incluidas dentro del entorno, y también se han creado otras necesarias para el desarrollo del trabajo.

2.6 Entorno socioeconómico

Dentro del entorno socioeconómico, uno de los campos que más importancia está dando a la detección de objetos es el automovilístico. Los accidentes de tráfico suponen una de las principales causas de fallecimiento, estando situados entre las diez primeras a nivel mundial. Por ello, los fabricantes están realizando fuertes inversiones en el desarrollo de *sistemas de ayuda al conductor* [15], (detección de proximidad con otros vehículos, salidas de carril, cansancio del conductor...). La mayoría de estos sistemas basan su tecnología en la detección de objetos, ya sea para indicar una posición, para detectar una posible colisión o la cara del conductor.

En el ámbito de la seguridad la detección de objetos también desempeña un papel muy importante. Además de seleccionar un objeto, se puede clasificar atendiendo a su valor, con vistas a poder activar una alarma en el momento que desaparezca [16], por lo que cuando desaparezca de la imagen se activa de forma automática una alarma o hacer un seguimiento del objeto. Por otro lado, es una gran solución para el conteo de personas o vehículos en una zona determinada, esta función es muy útil en ejerciendo desempeñando en supermercados, discotecas o parkings.

También ha de destacarse que en estos momentos un gran número de aplicaciones móviles requieren de reconocimiento facial. Quizás el ejemplo más claro sea Instagram [17], donde tienes la opción de aplicar un filtro facial, las famosas orejas de perro. Para ello se procesa el contrate de áreas por colores, seguido del cálculo de la proporción del rostro mediante varios modelos de ensayo. Por eso, si el filtro detecta una cara en el fondo, esto se debe a una mala iluminación del entorno y una confusión en el contraste de áreas.

Por último, pero no por ello menos importante, el campo que más repercusión está teniendo en estos momentos y que más se está desarrollando es el de los coches autónomos, donde Google [18] lleva la delantera. Los investigadores aseguran que uno de los mayores retos es su incorporación al mundo real, ya que los seres humanos no se

comportan como las simulaciones; efectivamente no se está teniendo en cuenta una componente subjetiva debido a la diferente manera de conducir de cada usuario, los conductores reales no son perfectos.

3. DISEÑO DE LA SOLUCIÓN TÉCNICA

3.1 Introducción y diagrama de bloques del sistema

El diseño del sistema está basado en los tres primeros objetivos específicos del proyecto, introducidos en la sección 1.2. Podemos distinguir tres bloques en el sistema propuesto: muestreo de ventanas (sección 3.3), extracción de características (sección 3.4) y localización de objetos (sección 3.5). Además, podemos diferenciar dos fases de ejecución del sistema: una fase de entrenamiento y otra de localización de objetos:

- Fase de entrenamiento: Partiendo de una base de datos de imágenes de objetos anotados mediante ventanas o *bounding boxes* (BB), el sistema tiene en cuenta estos en primer lugar como muestras positivas y, a continuación, muestrea ventanas negativas, pertenecientes al fondo. Todas estas ventanas son representadas a partir de un conjunto de características, con vistas a entrenar un clasificador que permita distinguir entre ventanas que contienen objeto y ventanas pertenecientes a fondo.
- Fase de localización de objetos: Una vez entrenado el modelo, el sistema muestreará ahora ventanas sobre imágenes en las que desconocemos la localización de los objetos. Estas ventanas son una vez más representadas por el mismo conjunto de características utilizado para describir las ventanas en la fase de entrenamiento. En última instancia, el modelo de clasificación entrenado determinará una probabilidad de existencia de objeto en cada una de las ventanas, ejecutando la tarea de localización de objetos sobre las imágenes, objetivo final de este proyecto.

En el proyecto se considerarán tres modelos de clasificación basados en SVMs. El primero de ellos recibirá a su entrada tres características: densidad de bordes (DB), contraste de color (CC) y superpíxeles transzonales (ST). El segundo tendrá como entrada una característica multidimensional, basada en la norma del gradiente de la imagen (NG). El tercer clasificador, finalmente, resultará de una fusión sencilla de las salidas de los dos primeros clasificadores, calculando la media de las probabilidades de objeto obtenidas para cada uno de los dos primeros clasificadores.

En la Figura 10 se incluye un diagrama de representación de las fases de muestreo de ventanas candidatas y extracción de características del sistema.

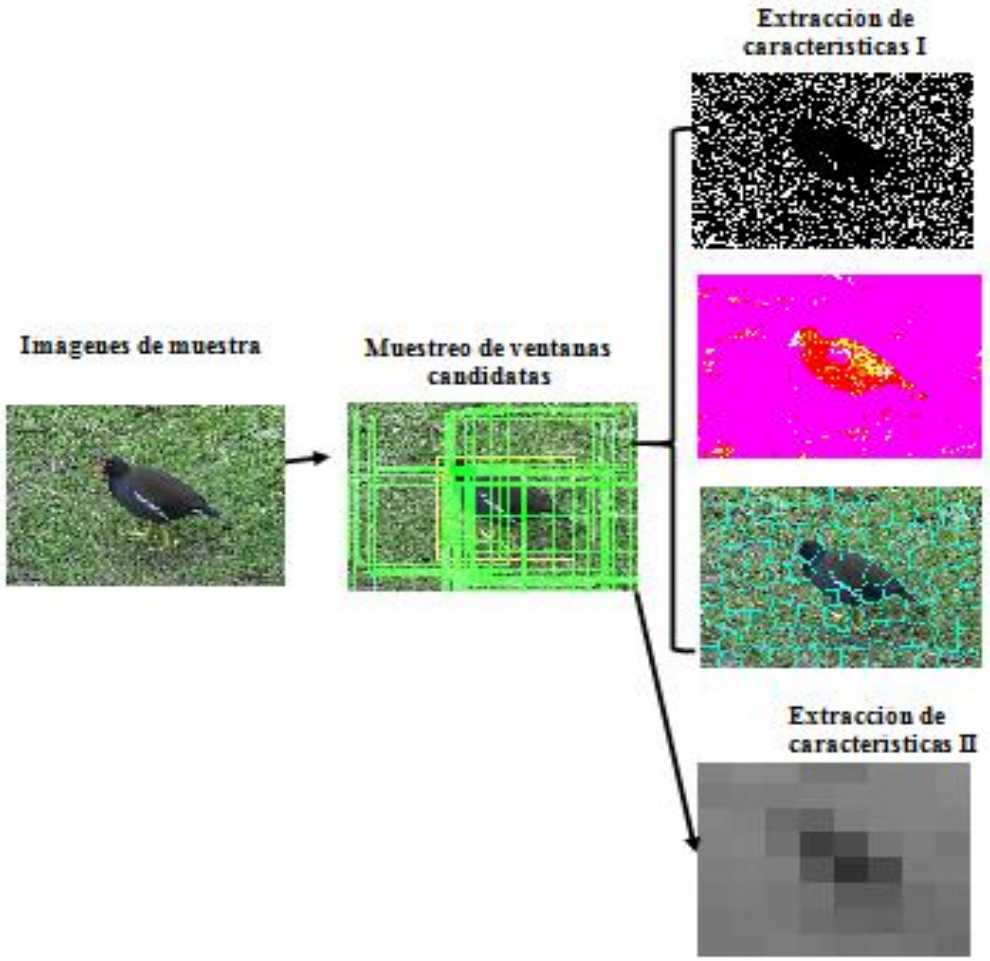


Figura 10. Esquema de las fases de muestreo de ventanas candidatas y extracción de características del sistema

3.2 Herramientas utilizadas

En este punto se describen todas las herramientas utilizadas para la elaboración del proyecto. Se tienen en cuenta las herramientas de software, las librerías de programación, así como el código externo o webs de consulta utilizadas.

- Como entorno de desarrollo se ha utilizado MATLAB R2018a para implementar las funciones necesarias para la realización del trabajo. Se ha trabajado con la última versión, que contiene algunas funciones más avanzadas en el ámbito del tratamiento de la imagen, como la utilizada para la división de las imágenes en superpíxeles. Se seleccionó este entorno porque no sólo permite la programación y ejecución del código, sino también la visualización de las imágenes y la depuración de las técnicas empleadas.

- Se ha utilizado una función externa para el cálculo de la distancia Chi-cuadrado de dos histogramas, la función es de libre disposición y descarga gratuita desde la comunidad de MathWorks [19].

- Para el desarrollo del trabajo se ha empleado el ordenador del alumno y, en ocasiones, algún ordenador de los laboratorios de la universidad.

3.3 Muestreo de ventanas candidatas basado en histogramas

El primer reto al que se tuvo que hacer frente fue la generación de ventanas candidatas a contener un objeto en la imagen. Como ya se ha mencionado con anterioridad, el método más empleado en los primeros localizadores de objetos en el estado del arte es el de *ventana deslizante.* Su principio de funcionamiento es muy sencillo: se generan ventanas con cierto solapamiento, atendiendo a todas las localizaciones espaciales de la imagen, y con distintos tamaños, para buscar dentro de ellas objetos a diferentes escalas.

Este método no es óptimo pues han de generarse una gran cantidad de ventanas para cubrir la imagen atendiendo a las diferentes escalas, lo que hace que su coste computacional sea muy elevado.

En nuestro caso, hemos optado por un procedimiento de muestreo diferente, más eficiente. Como solución a esta etapa del sistema adoptaremos una técnica basada en un histograma de localizaciones y relaciones de aspecto a priori. Partiendo de un conjunto de imágenes de entrenamiento lo suficientemente genérico y representativo de una gran variedad de objetos en fotografías (vacas, trenes, aviones, etc.), se puede construir un histograma teniendo en cuenta la localización *Ground-Truth* (GT) de estos objetos, determinada por sus correspondientes BBs.

Dado un objeto, su BB está definida por cuatro valores:

$$[x, y, W, H]$$

Los valores de las coordenadas x, y se corresponden con su coordenada superior izquierda; W es el ancho de la imagen; y H es el alto de la imagen.

En el histograma que vamos a construir, las coordenadas (x, y) determinarán la localización del objeto en la imagen, atendiendo a su sector correspondiente de acuerdo la "regla de los tercios" [10]. introducida en el Capítulo 2, mientras que H y W servirán para definir, dado un sector, un histograma bidimensional de relaciones de aspecto. De este modo, en la fase de localización de objetos, las ventanas se muestrean en las zonas de la imagen en las que es estadísticamente más probable que se encuentre el objeto, y con el tamaño que es más probable que se dé.

El método para determinar en qué sector o sectores se encuentra el área del objeto consistió en crear una máscara con la misma superficie que el sector con la que determinar en que posición se encuentran las esquinas de la BB del objeto.

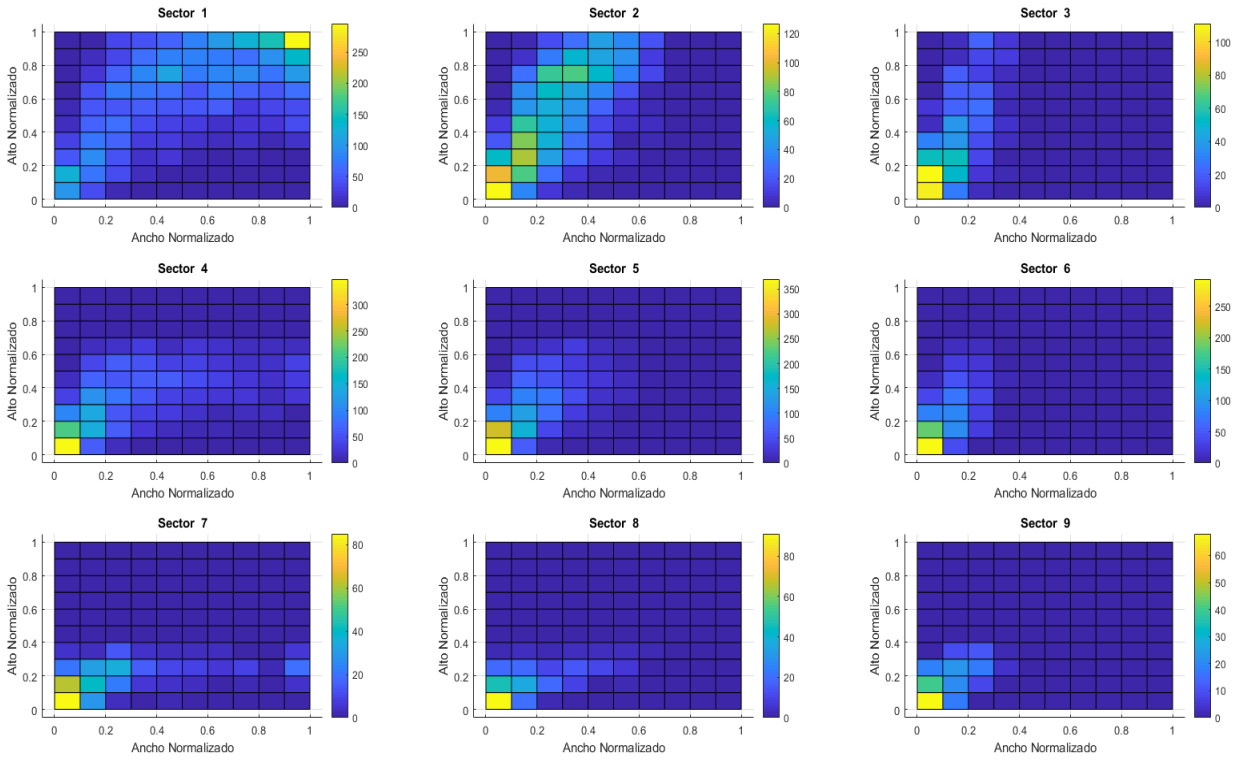
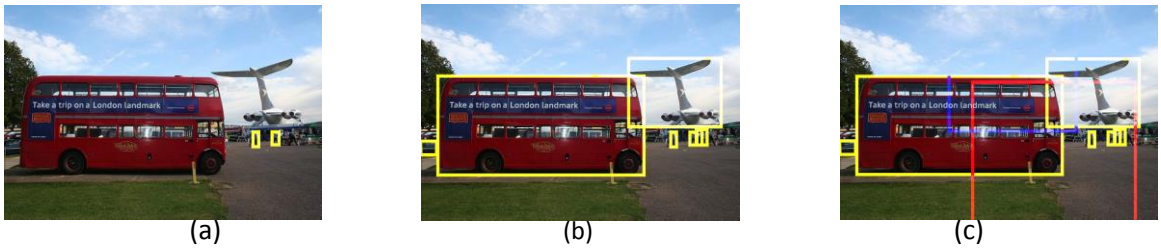


Figura 11. Histogramas bidimensionales de relaciones de aspecto para cada una de las localizaciones de imagen definidas por los sectores considerados en la "regla de los tercios" [10].

Se toma el sector donde haya una concordancia con la BB, y nos quedamos con la información de la esquina superior izquierda, que representa el punto de partida del objeto en la imagen.



(a) (b) (c)

Figura 12. Ejemplo de fotografía de la base de datos PASCAL VOC [21]

Finalmente se usan los datos recopilados en primer lugar para cada uno de los sectores para obtener un histograma general. El proceso para lograrlo, por tanto, consiste en realizar un histograma individual para cada recinto, donde recibe como parámetros el ancho y alto normalizados de todas las BB en esa región. Una vez se tienen los nueve histogramas (uno por cada sector de la imagen), se concatenan en un histograma general donde se recopila la información de todas las localizaciones.

El histograma formado ofrece una representación de la distribución de la frecuencia de las zonas donde es más probable encontrar los objetos con respecto de la imagen. Este histograma, representativo de una muestra amplia de objetos, además, sirve de punto de partida para tomar decisiones sobre dónde generar las ventanas candidatas a objeto en la fase de localización.

Para seleccionar los parámetros correspondientes a cada una de las ventanas candidatas ($[x, y, W, H]$), se muestrea aleatoriamente de manera probabilística sobre el histograma construido las tres variables [sector, ancho, alto]. Dado el sector, se muestrea aleatoriamente la posición de su esquina superior izquierda (x, y). La relación de aspecto de la ventana, dada por las dimensiones alto y ancho normalizadas, permite definir la ventana en la fotografía.

Un ejemplo donde se aplica la técnica de muestreo de ventanas candidatas por histograma es la figura 13, donde se puede observar en la primera fotografía a un hombre en el centro de la imagen, y una serie de ventanas candidatas que intentan localizarlo. Algunas no contienen ni un solo píxel del objeto en su área, pero otras sí parecen localizarlo, además con un tamaño aproximado. En la imagen de la derecha se ve una bicicleta, de mayor tamaño que el anterior, y posiblemente también mayor que la media, por lo que algunas de las ventanas de muestra no la delimitan por completo. Este es el motivo de que se muestree con distintos tamaños de ventana, de forma que no solo se localice el objeto sino también con el borde más adecuado al del elemento en cuestión.

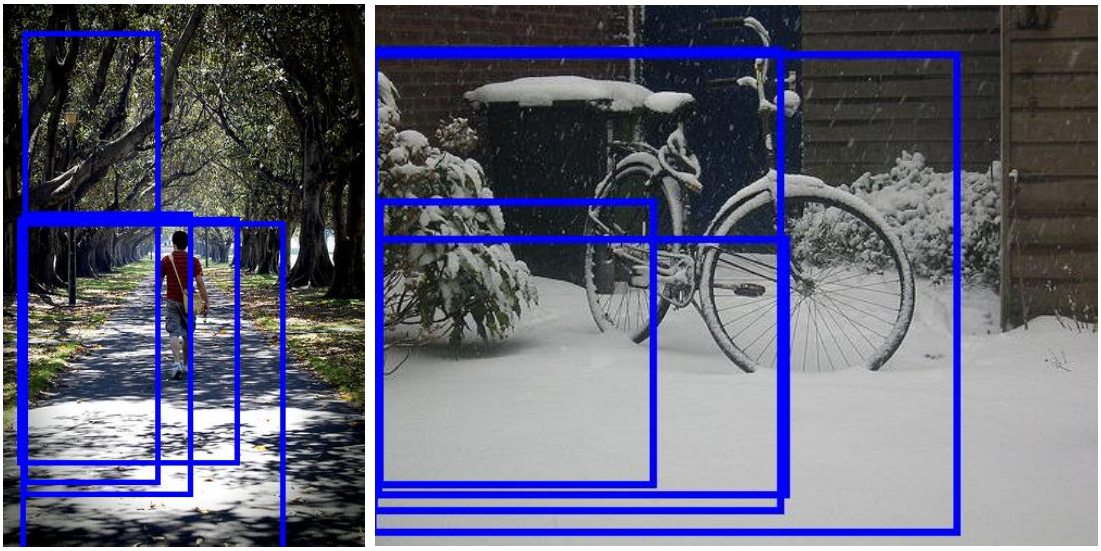


Figura 13. Ejemplo de fotografías de la base de datos PASCAL VOC [21] en las que se muestrean ventanas candidatas a objeto con el procedimiento basado en un histograma de localizaciones y relaciones de aspecto propuesto

Para hacer más fiel el muestreo de ventanas al entorno real de trabajo, se generó un pequeño valor aleatorio que representase ruido, uno que afectase a las coordenadas de partida y otro para las dimensiones de la ventana. Con esto se evita en cierta medida la reiteración en las coordenadas para un mismo sector, lo que hará que se distribuyan de manera más uniforme. Además, permite muestrear ventanas con relaciones de aspecto diferentes a las representadas en las posiciones discretas del histograma.

Como último paso a este procedimiento se superpone la ventana a la imagen para extraer los descriptores correspondientes el área de análisis.

Este método se fundamenta siempre en la misma base, utilizar los valores obtenidos en el histograma para la generación de ventanas candidatas, pero hay que puntualizar una serie de diferencias que se dan entre la fase de entrenamiento y la fase de evaluación. Al tratarse de etapas diferentes del sistema no se pueden emplear exactamente el mismo

procedimiento en ambos casos: en la fase de entrenamiento se dispone de las BBs que localizan los objetos en la imagen, las cuales constituirán el conjunto de muestras positivas con el que se entrenará el clasificador. En cambio, la única información que se dispone en la fase de evaluación acerca de la localización de los objetos en las fotografías vienen data por el histograma que hemos construido a partir de las ventanas positivas o con objeto en la fase de entrenamiento.

3.3.1 Fase de entrenamiento

En la fase de entrenamiento, podemos distinguir entre ventanas positivas y negativas:

- El conjunto de ventanas positivas estará formado por aquellas que contienen objetos, correspondientes a las BBs anotadas en la base de datos. Estas ventanas serán las únicas en el conjunto de entrenamiento cuya etiqueta tendrá el valor de 1 (significa que son una ventana que contienen al objeto por completo).

- Por otro lado, para generar el conjunto de ventanas negativas, se muestrearán 400 ventanas candidatas para cada imagen a partir de la técnica del histograma antes descrita, en localizaciones de la imagen donde no hay objeto. Estas muestras negativas se darán en zonas en las que, a priori, es muy probable que existan objetos; por tanto, son muestras negativas difíciles de clasificar ("*hard negatives*"), con las que esperamos obtener modelos de clasificación más robustos. Sobre estas 400 ventanas candidatas, aplicamos un primer filtrado para reducir el número de ventanas. La técnica consistirá en superponer una a una las ventanas negativas candidatas con las BBs positivas. Las ventanas negativas seleccionadas serán aquellas cuyo solapamiento con las ventanas positivas sea de un 0.2 a un 0.5 (tanto pro uno). Cuando la imagen contenga más de un objeto, todos los objetos han de cumplir el criterio anterior para que la ventana sea seleccionada como negativa.

Para prevenir que el número de ventanas candidatas siga siendo demasiado elevado se realiza un segundo filtrado: en caso de que el número de muestras sea mayor de 40 (número que se ha determinado como valor óptimo de máximo volumen de ventanas candidatas para el conjunto de entrenamiento), se ordenan las ventanas en orden ascendente según el área de solapamiento, y se seleccionan 40 muestras equiespaciadas. que conformarán el conjunto final de ventanas negativas para cada imagen en la fase de entrenamiento. A estas ventanas se les asignará como etiqueta el valor de 0.

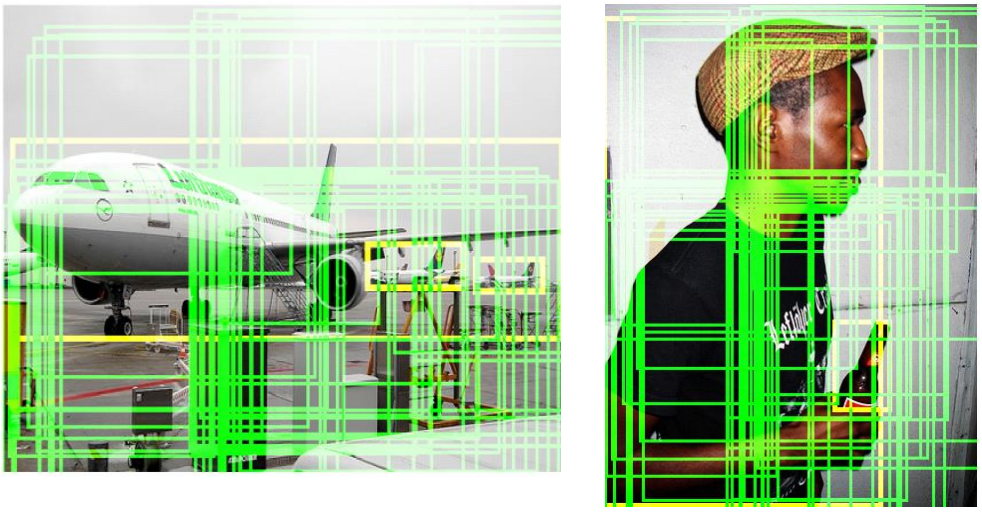


Figura 14. Fotografías con objetos, la BB (rectángulo amarillo), y las ventanas generadas por el muestreo de histograma (rectángulos verdes).

3.3.2 Fase de localización de objetos

El muestreo de ventanas candidatas tanto en la fase de localización del objeto, como en la fase de validación de parámetros del modelo, se basará en el histograma de localizaciones y relaciones de aspecto obtenido a partir de las BBs de objetos en imágenes del conjunto de entrenamiento. Partiendo del mismo, se muestreará aleatoriamente de manera probabilística un determinado número de ventanas, y se procederá a evaluar el rendimiento del sistema atendiendo a las medidas de evaluación introducidas en el capítulo siguiente.

3.4 Extracción de características

3.4.1 Densidad de Bordes (DB)

La implementación de este descriptor se basa en las ideas descritas en los artículos [6,7], donde el objetivo es lograr localizar un objeto a través de los bordes de su contorno. Tal y como se introdujo en el capítulo anterior, un objeto se diferencia de su entorno a partir de su estructura delimitada y bien definida. Esto sin embargo en ocasiones no es del todo cierto, debido a que los cambios de iluminación en la imagen pueden interferir en la correcta apreciación del contorno.

El proceso tiene en cuenta los píxeles que componen el interior de un "anillo interno", Int(w, θ), que se obtiene reduciendo de manera proporcional la ventana candidata, un cuarto del total del ancho de la imagen en cada lado. Esta reducción se ha determinado de manera empírica. Para la aplicación de esta técnica es necesario que la fotografía sea en escala de grises, por lo que previamente se realiza una transformación a este espacio.

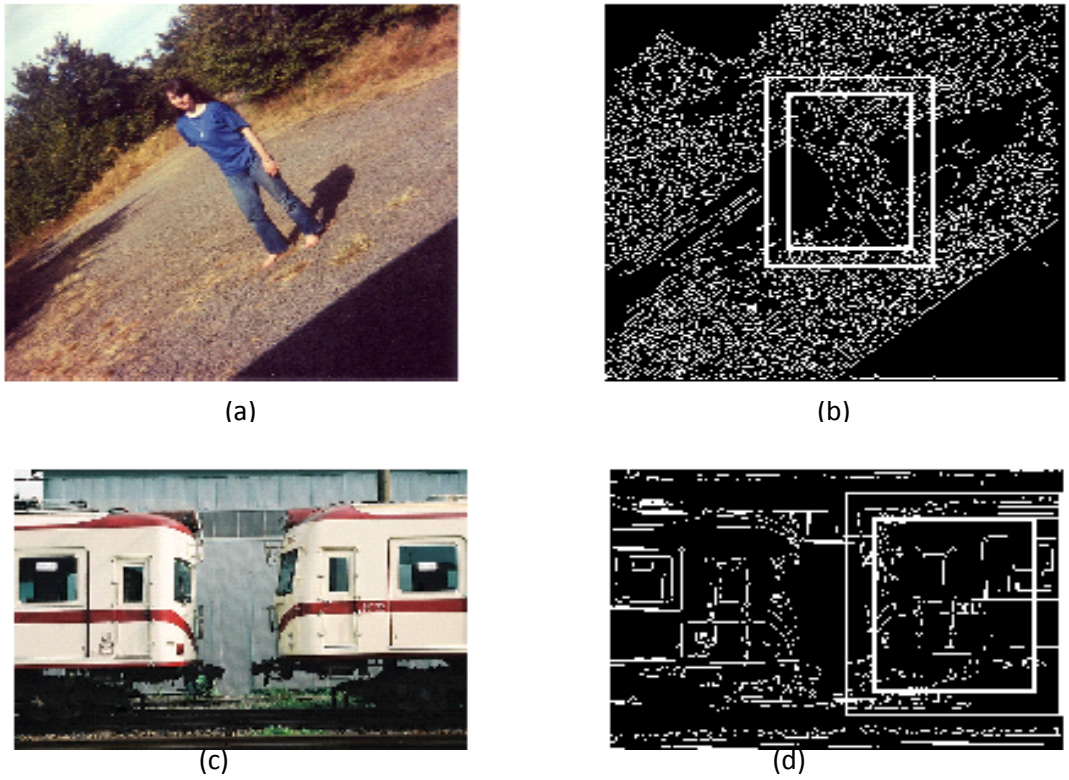


Figura 15. Ejemplos de DB donde (a) y (c) son las fotografías y (b) y (d) los respectivos resultados de aplicar la técnica.

A continuación, se calcula la densidad de bordes en el anillo interno. Para ello se ha usado el detector de Canny descrito en el capítulo anterior (sección 3.1). El resultado es la suma de los píxeles que han sido definidos como bordes tras la aplicación del método, dividida por el perímetro del anillo interno, Per($Int(w, \theta_{ED})$).

$$DB(w, \theta_{DB}) = \frac{\sum_{p \in Int(w, \emptyset_{DB})} I_{DB}(p)}{Per(Int(w, \theta_{DB}))} \qquad (3.1)$$

3.4.2 Contraste de Color

Este descriptor de Contraste de Color (CC) también se fundamenta en los estudios previos recogidos en [6,7]. Su objetivo es diferenciar el objeto del paisaje, o del fondo que le rodea, a partir del contraste medido en el espacio de color Lab.

Para ello, se construye un anillo, ahora externo, alrededor de la ventana candidata, $Ext(w, \theta)$,. En efecto, el margen que se aplique aumentará el tamaño de la ventana, de forma opuesta a como se hizo en el caso anterior.

Se calcula la diferencia de contrastes como la distancia de Chi-cuadrado $\times^2$ entre el conjunto de histogramas Lab del anillo exterior y conjunto de histogramas de la ventana candidata. Para cada componente se construye un histograma con un número de bins diferente: 8 para la componente *L*, 16 para la componente *a* y 16 para la componente *b*. A continuación, se concatenan, por un lado, los histogramas Lab correspondientes al anillo externo (entorno) y, por otro, aquellos representativos de la ventana candidata,

formando los histogramas $h(w), h\left(Ext(w, \theta_{CC})\right)$, y se calcula como descriptor la distancia entre los mismos:

$$CC(w, \theta_{CC}) = \times^2 \left(h(w), h\left(Ext(w, \theta_{CC})\right)\right) \quad (3.2)$$

La posición de la ventana puede estar cerca o en algún extremo de la fotografía, lo que supondría que al establecer el anillo exterior algunos de sus puntos no representasen valores de la imagen. Para este tipo de situaciones se ha empleado la función de MATLAB *padarray*, cuya función es ampliar la matriz dando valores a los píxeles que no estén dentro del margen, pero sí dentro del anillo exterior. Para hacerlo se aproximan lo máximo posible esos valores: se define el parámetro de la función *padarray* encargado de seleccionar el algoritmo que definirá los nuevos píxeles como *symmetric*, lo que hace que el programa actúe como un espejo a la hora de estimar los nuevos valores. De esta forma se obtiene una representación más fiel al conjunto completo que si solo se reprodujese el píxel vecino o se hiciera la media entre un pequeño grupo de píxeles.

Tal y como se muestra en la Figura 18, para que los histogramas sean representativos de la ventana candidata y del anillo exterior y no tener que recurrir a ninguna aproximación se crean dos máscaras específicas para cada caso: una es la máscara interna que representa los píxeles que forman el conjunto de la ventana candidata, y la otra es la máscara exterior que representa al conjunto de datos del anillo. Estas máscaras se aplican antes de calcular el histograma, y se utiliza un filtro para descartar los valores que no se deseen. Es muy sencillo de utilizar ya que la máscara habrá anulado los valores que no sean necesarios para ese histograma. Luego se repite el mismo proceso con la máscara que no haya sido utilizada, para de este modo tener una representación de todos los datos.

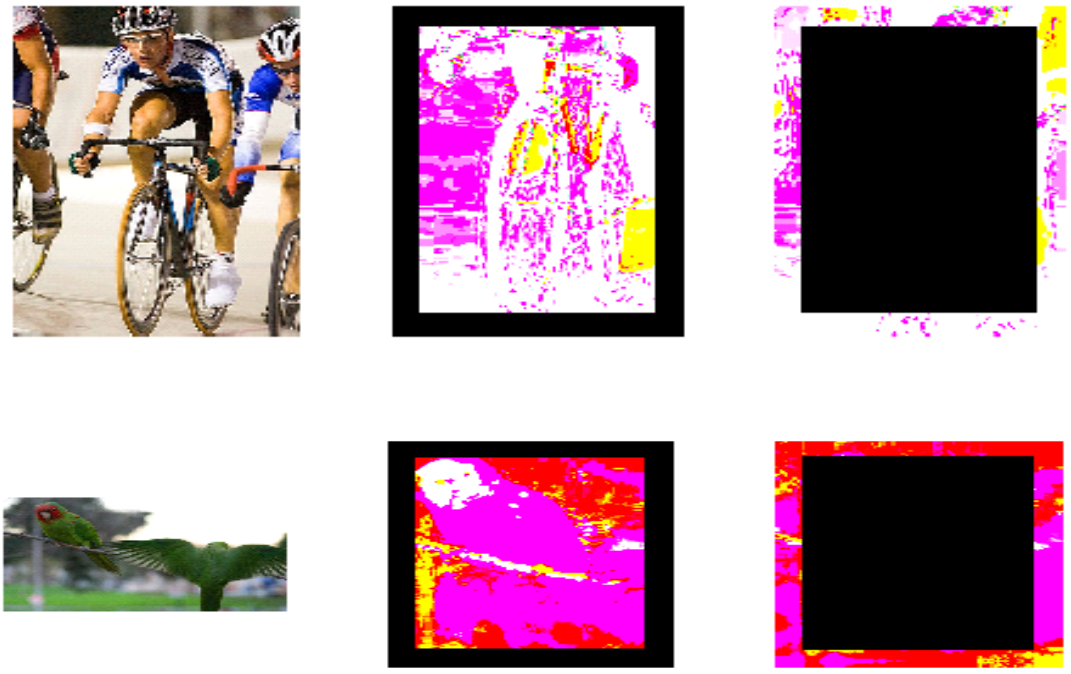


Figura 16. De izquierda a derecha se representan la imagen de muestra, la imagen pasada a LAB con la máscara exterior y la misma imagen LAB, pero ahora con la máscara interior

3.4.3 Superpíxeles Transzonales (ST)

Es la última característica que se introduce al primer clasificador. Su objetivo, al igual que el del descriptor DB, es caracterizar un límite cerrado. Para ello, en esta ocasión se utiliza una técnica diferente [6], basada en el empleo de los superpíxeles. Como ya se ha descrito anteriormente, los superpíxeles segmentan la imagen en pequeñas regiones que tienen aspectos en común.

El principio fundamental en el que se basa la técnica es que todos los píxeles pertenecientes a superpíxeles forman parte de un mismo objeto y que dicho elemento está divido a su vez en un conjunto finito de superpíxeles, por lo que el objetivo es la localización de esos superpíxeles que en conjunto forman el objeto. En base a esta propiedad, se llega a la conclusión de que, si una ventana candidata cubre un objeto por completo, la mayoría de superpíxeles contenidos se encontraran completos, es decir, no habrá píxeles que pertenezcan a su conjunto y estén fuera de los límites de la ventana.

En caso contrario, si la ventana no presenta ningún objeto, la superficie estará compuesta por superpíxeles que tienen parte de sus píxeles dentro de la ventana y otra parte fuera del recinto. El descriptor de Superpíxeles Transzonales (ST) tiene en cuenta, para cada superpíxel s que traspasa la frontera definida entre la ventana candidata w y el fondo de la imagen, el número mínimo de píxeles que caen en uno de estos dos lados: dentro $|s \cap w|$ o fuera $|s \setminus w|$ de la ventana. Aquellos superpíxeles que no traspasan la frontera no se tienen en cuenta a la hora de calcular este descriptor.

$$ST(w, \theta_{SS}) = 1 - \sum_{s \in S(\theta_{SS})} \frac{\min(|s \setminus w|, |s \cap w|)}{|w|} \quad , \qquad (3.3)$$

donde |.| se utiliza para indicar el área de la región correspondiente, en número de superpíxeles.

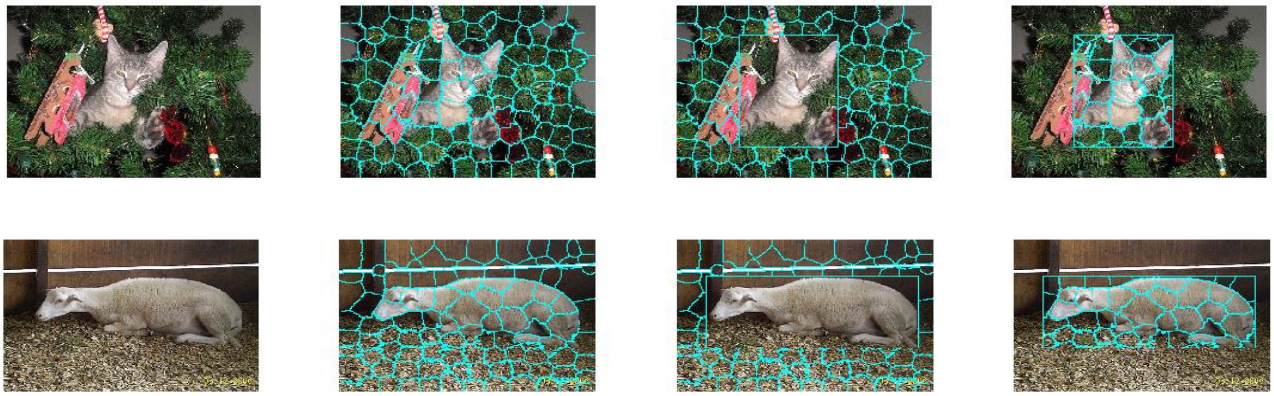


Figura 17. Imagen de muestra, la imagen dividida por superpíxeles, los superpíxeles que forman el marco exterior y los superpíxeles que forman la ventana candidata.

Para desarrollar esta técnica también ha sido necesario, como en el caso previo, el uso de máscaras, como se muestra en la Figura 17. En este caso ha sido necesaria la creación de tres mascaras para acotar los índices de los superpíxeles, ya que todos los píxeles de un superpíxel tienen el mismo índice.

La primera máscara se forma para el marco que forma la ventana, para conocer el índice de los superpíxeles que forman parte de esta zona. La segunda de las máscaras es la del marco, de la ventana hacía fuera, para obtener los índices de los superpíxeles que pueden formar parte del entorno. En tercer lugar, la máscara del interior de la ventana candidata, con la que se obtienen los índices restantes para así poder aplicar la fórmula 3.3. Se deben eliminar previamente los píxeles cuyos índices no coincidan con los del marco, que no contribuyen al descriptor, como ya se expuso anteriormente.

3.4.4 Descriptor de norma del gradiente de la imagen (NG)

Este descriptor se utilizará como entrada a un clasificador diferente al usado para los tres primeros descriptores. El desarrollo de esta técnica se fundamenta en que los objetos con límite cerrado y definido se pueden diferenciar del entorno por los valores de la norma de su gradiente [12]. Por ello, dada una imagen, se genera su correspondiente imagen de módulo o norma del gradiente. A continuación, cada ventana candidata muestreada sobre esta imagen de módulo de gradiente es redimensionada a tamaño 8x8, obteniendo un descriptor 8x8=64-dimensional de la misma.

Figura 18. Imagen, imagen con la BB que encuadra al objeto y aplicar la técnica BING a la BB

Figura 19. Imagen con ventana candidata y aplicar la técnica BING a la ventana candidata

Este descriptor es sencillo, pero no por ello menos efectivo que los anteriores. Como se aprecia en la Figura 20, cuando la ventana deslizante delimita correctamente el objeto, al convertirla a 64 píxeles hay un elemento que se diferencia del resto. En el caso de la Figura 18, que tiene la ventana deslizante en una zona donde la mayoría del área pertenece al fondo, en este caso cielo, al convertirla a 64 píxeles la mayoría de píxeles tendrán el mismo valor, y los únicos con un valor diferente son aquellos que formen parte del objeto, en el caso de este ejemplo la parte delantera del avión.

En ambos casos se parte de la misma foto, pero se obtienen resultados bastante diferentes debido a la posición de la ventana, por lo que a primera vista se puede decir que es una buena técnica para la representación de ventanas que contienen objetos.

3.5 Clasificador: Máquinas de Vectores Soporte (SVM)

La etapa de clasificación permite separar las ventanas candidatas en dos clases: objeto y fondo. Para ello se utilizará un clasificar de Máquinas de Vectores Soporte introducido en el capítulo anterior con kernel lineal.

El clasificador está compuesto por tres fases: entrenamiento, validación y test, que siempre deben ejecutarse en este mismo orden. El objetivo del clasificador al finalizar este proceso es que sea capaz de proporcionar la etiqueta adecuada con la mayor precisión a cada ventana candidata de un conjunto de imágenes en el que se quiere localizar objetos. El clasificador no es perfecto, su nivel de acierto no es del 100%, pero si se ajusta adecuadamente en las fases previas pueden obtenerse unas buenas prestaciones que cumplan unos ciertos requisitos.

En todas las fases las imágenes tienen un número de ventanas candidatas, y siempre el número equivalente de etiquetas asociadas a las mismas. Las etiquetas informan acerca de si esa ventana contiene un objeto (el valor de la etiqueta es 1) o, si, por el contrario, no hay objeto o no se cumple con los criterios mínimos (detección de sólo partes de objetos, por ejemplo, el asa de una taza), en cuyo caso la etiqueta tendrá valor 0.

3.5.1 Fase de entrenamiento

La fase de entrenamiento tiene como objetivo encontrar el hiperplano óptimo, que divida las ventanas muestreadas sobre las imágenes en dos clases. Como en este caso se trabaja con dos clases se trata de una clasificación binaria, entre objeto y fondo

La fase en entrenamiento es el primer paso del clasificador. En primer lugar, se selecciona un conjunto de datos que sirven para entrenar el modelo de predicción. El sistema irá ajustando el modelo según los datos de entrenamiento.

La máquina recibirá por tanto muestras $[x_{1,}x_{2,}, x_{3,}, \dots\ x_n]$ con sus etiquetas correspondientes $[y_{1,}y_{2,}, y_{3,}, \dots\ y_n]$, siendo éstas 1 cuando la muestra sea de objeto, y 0 cuando la muestra sea de fondo.

3.5.2 Fase de validación

La validación se emplea para determinar la configuración óptima que debe tener el sistema, para después poder entrenar con el conjunto completo de muestras. Para ello se van seleccionando los distintos valores de los costes posibles, y se aplica la validación cruzada que se explica a continuación.

Los datos se trabajan según la técnica de validación cruzada: el conjunto de datos se divide en 5 grupos, 4 son usados para entrenar e ir ajustando el modelo y el restante se usa como grupo de validación. Estos grupos irán rotando de forma que todos los grupos deben servir como conjunto de validación mientras el resto sirven de entrenamiento.

A continuación, se muestra un esquema que representa el proceso de validación cruzada, de donde se parten de los datos de entrenamiento:

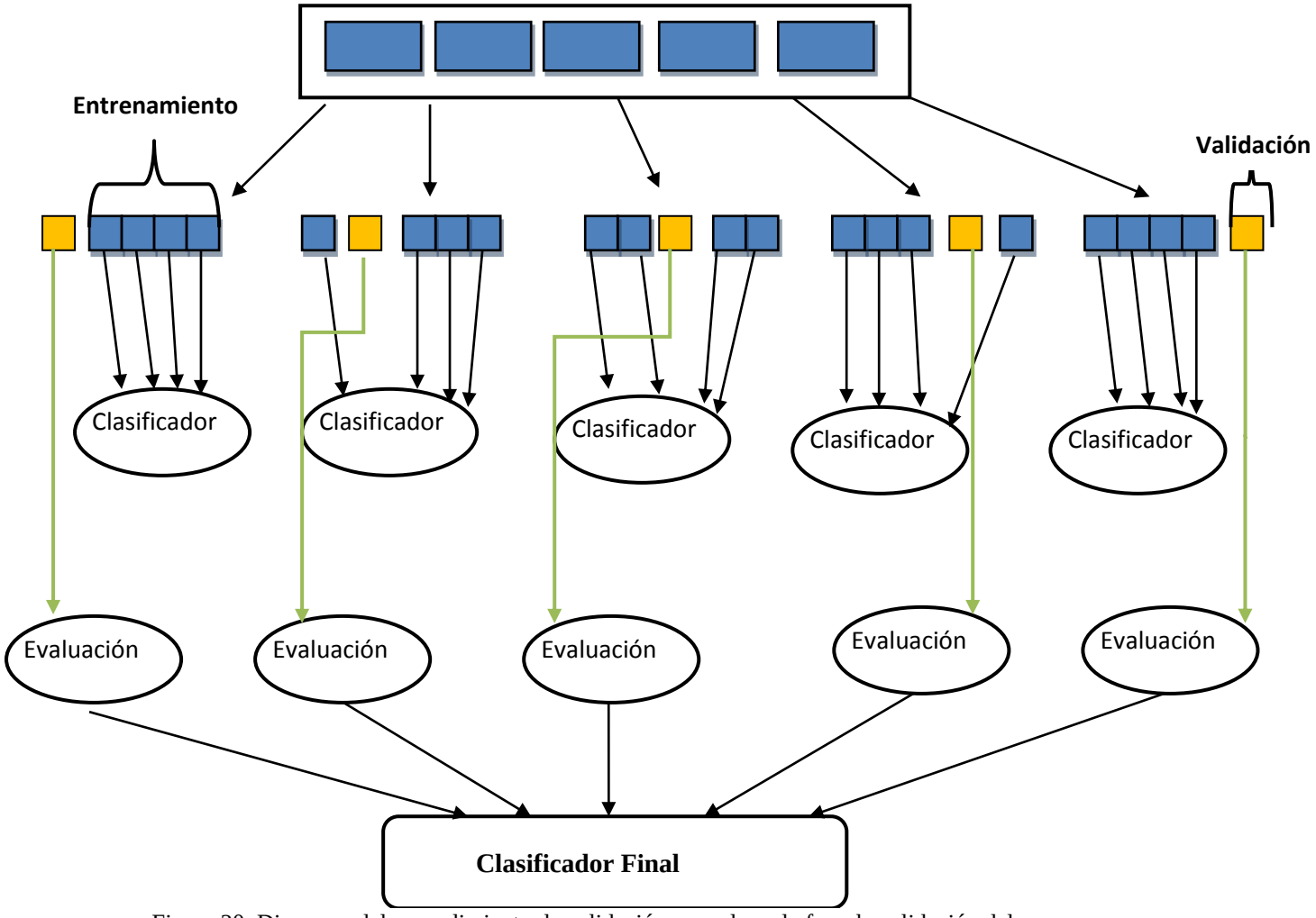


Figura 20. Diagrama del procedimiento de validación cruzada en la fase de validación del sistema

Para obtener el mejor rendimiento posible los datos usados deben tener un amplio número de muestras y que estas sean representativas de todo el conjunto, no sólo del de entrenamiento, ya que si el conjunto es muy específico se complicará la toma de decisiones en la validación y posteriores pruebas. Por ejemplo, si solo se entrena con fotos de trenes cuando se quiera clasificar una foto con un árbol lo más seguro es que su

resultado sea como no objeto, esto es porque la muestra de entrenamiento no era representativa de todo el conjunto de objetos genéricos.

3.5.3 Fase de test

Cuando ya se tiene un clasificador entrenado y se han definido los pesos óptimos, para lograr la mejor catalogación de los datos posible. Se debe probar el modelo en un grupo de muestreo del que no se conozcan las etiquetas.

En esta fase únicamente se evaluará al clasificador. Se le pasan un conjunto de imágenes y se evalúan los resultados dados por este en ventanas muestreadas por las mismas a partir del histograma de localizaciones y relaciones de aspecto. Las etiquetas que se asignen a las muestras dependerá del valor de score obtenido. En los casos donde el score tenga un valor superior a 0.5, recibirán un valor de 1, etiquetadas como que contienen objeto. Mientras que, en el caso contrario, aquellas muestras cuyo score sea menor a 0.5, se les asignará un valor de 0, etiquetadas como que contienen fondo.

4. RESULTADOS Y EVALUACIÓN

4.1 Introducción

A continuación, se evaluarán 3 sistemas de localización de objetos basados en clasificadores SVMs de tipo lineal, que toman como entrada las características descritas en el capítulo anterior. Los sistemas son los que se representan en los diagramas siguientes:

Figura 21. Diagrama del primer clasificador SVM propuesto

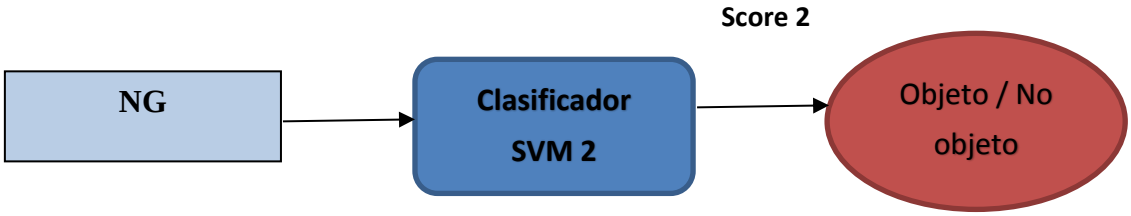


Figura 22. Diagrama del segundo clasificador SVM propuesto.

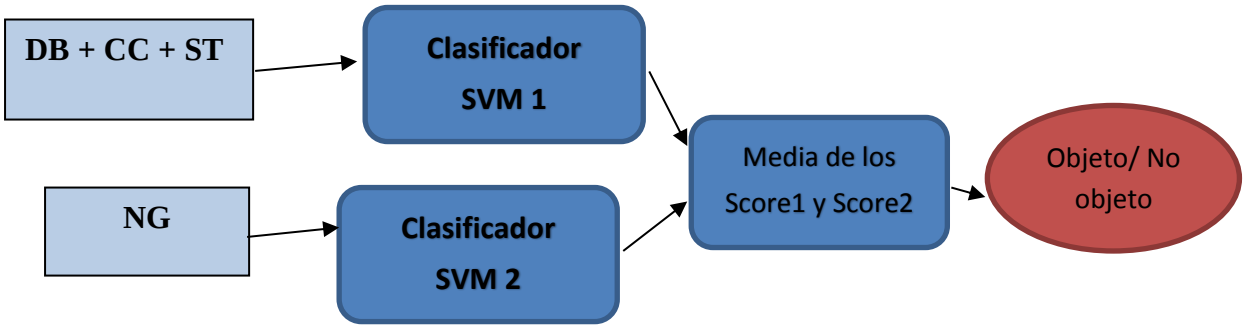


Figura 23. Diagrama del tercer clasificador SVM propuesto, fusión de los dos primeros

- La Figura 21 muestra el diagrama del primer clasificador, que recibe las tres primeras características descritas en el capítulo anterior: densidad de bordes (DB), contraste de color (CC) y superpíxeles transzonales (ST), todas ellas valores numéricos. En la fase de localización del sistema, el modelo de clasificación obtenido en la fase de entrenamiento determinará, a su salida, para cada una de las ventanas muestreadas en una imagen nueva (no vista en la fase de entrenamiento), si contiene un objeto o pertenece al fondo.

- El segundo clasificador está representado en la Figura 22. El clasificador funciona de la misma manera que el primero, pero, a diferencia de éste, recibe a su entrada una única característica, la norma del gradiente, 64-dimensional.

- Por último, el tercer clasificador resulta de la fusión sencilla de los dos primeros clasificadores. Su salida se consigue realizando la media de los scores obtenidos para una misma ventana, en una misma imagen, para cada uno de los dos primeros.

Todos los sistemas se evaluarán atendiendo a los mismos criterios, y sobre los mismos conjuntos de imágenes de la base de datos PASCAL VOC [21], los cuales se describen en la sección siguiente. Para ello, se determinará, en primer lugar, en una fase de validación, el valor óptimo del parámetro C asociado a cada uno de los dos primeros clasificadores lineales. Utilizando el valor de C que proporcione unas mejores prestaciones, atendiendo a tres medidas de evaluación: accuracy, precisión y recall, se evaluará la eficiencia del sistema dependiendo del número de ventanas que se muestrean sobre las imágenes del conjunto de test, en la fase de localización de objetos del sistema.

Finalmente, se realizará un análisis de errores de los modelos propuestos, con vistas a valorar sus ventajas e inconvenientes. Esto servirá para introducir las líneas futuras de trabajo enumeradas en el capítulo siguiente, con vistas a mejorar las prestaciones del sistema desarrollado.

4.2 Base de datos

Para ofrecer una buena evaluación las prestaciones del sistema, se debe usar un conjunto de imágenes lo suficientemente grande, y que contenga objetos pertenecientes a una amplia variedad de categorías. Es por ello que, para presentar los resultados de este capítulo se ha usado la base de datos PASCAL VOC [21], que cuenta con 17.000 imágenes digitales clasificadas en 20 clases semánticas, en las cuales encontramos imágenes vacas, perros, ordenadores, bicicletas, etc.

Cada una de las imágenes contiene una estructura con información perteneciente a sus características. Citaremos algunos de los campos más relevantes y que se han usado:

- Ancho y alto de la imagen.

- Número de objetos que contiene la imagen.

- Coordenadas de la esquina superior izquierda de la BB del objeto u objetos.

- Ancho y alto de la BB del objeto u objetos.

La base de datos se presenta dividida en un conjunto de entrenamiento de 10.000 imágenes, de las cuales usamos 5.000 para la construcción del histograma de sectores y relaciones de aspecto mostrado en la Figura 12 del capítulo anterior. Debido a que procesar la base de datos completa en el equipo que se dispone sería computacionalmente muy costoso, sobre ese mismo grupo de imágenes de entrenamiento seleccionaremos un subconjunto de 1.000 imágenes para nuestros experimentos, que se usarán en la fase de entrenamiento del sistema. Este conjunto, además, será subdividido en cinco particiones en la fase de validación del sistema, con el objeto de obtener los parámetros C de las SVMs lineales que ofrezcan mejores prestaciones.

Por otro lado, el conjunto de imágenes de test de la base de datos contiene las 7.000 imágenes restantes, de las cuales utilizaremos también un subconjunto de 1000 imágenes para la fase de test o localización de objetos de nuestro sistema. En la figura 26 se observa la división completa de la base de datos descrita:

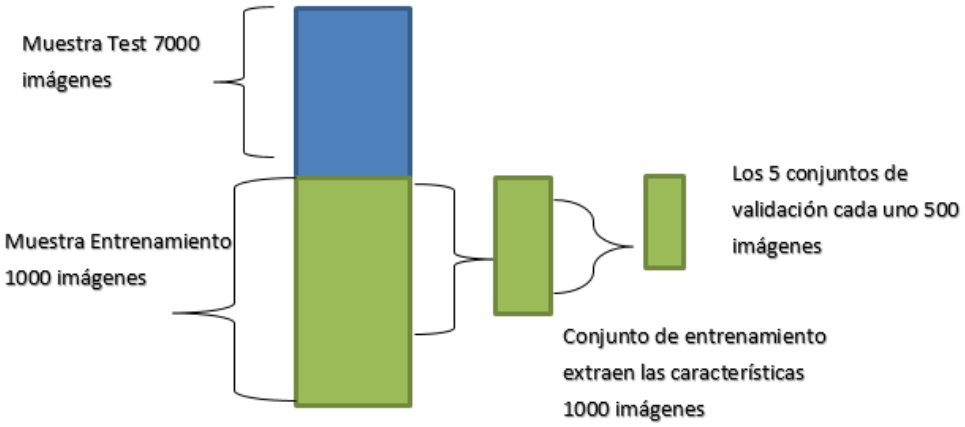


Figura 24. División base de datos PASCAL VOC [21]. Se indica el número de imágenes que forman los subconjuntos que se utilizarán en la fase de entrenamiento y localización de objetos para los experimentos realizados en el proyecto.

4.3 Medidas de evaluación

Para la evaluación del sistema, se tendrán en cuenta tres medidas clásicas, utilizadas habitualmente para determinar las prestaciones de sistemas de localización de objetos: accuracy, precision y recall. Los términos en los que se fundamentan estas medidas de evaluación son:

- ***Falso Positivo (FP)***: Muestras o, en nuestro caso, ventanas etiquetadas como positivas por el clasificador, pero cuyo verdadero valor es negativo. Es decir, ventanas en las que localizamos un objeto cuando realmente no lo hay.

- ***Verdadero Positivo (VP)***: Basándonos en el sistema de detección de objetos, objetivo de este proyecto, ventanas o BBs que contienen objeto y en las que se ha localizado éste de manera correcta.

- ***Falso Negativo (FN)***: También conocido habitualmente como muestra perdida, son aquellas BBs que contenían objetos en las que, sin embargo, no hemos sido capaz de localizarlos.

- ***Verdadero Negativo (VN)***: Equivalente al VP, pero para la otra clase, la que se establecido como negativa. Efectivamente, se corresponde con aquellas ventanas que no contenían objetos y en las que, de manera correcta, no hemos localizado uno.

La Tabla 2 resume estos conceptos para el caso concreto del proyecto, detectar objetos en BBs muestreadas sobre imágenes en las que no conocemos su localización.

Tabla 1. Tabla de que enfrenta valores reales con los valores obtenidos

		Resultado Real	
		Positivo	Negativo
Resultado Evaluación	Positivo	VP	FP
	Negativo	FN	VN

Una vez que se han aclarado estos conceptos se definen las medidas para la evaluación de los resultados. Estas medidas se han empleado tanto en la fase de validación como en la de test, ambas de localización de objetos, y son medidas que dependen mucho del valor que se establezca en dos parámetros al principio de estas fases. El primer parámetro a tener en cuenta es el número de muestras que se usan para evaluar, ya que un conjunto demasiado pequeño puede que no modifique o entrene de una forma suficientemente general al clasificador, lo que produciría un gran número de FN (la relación de muestras es aproximadamente de una ventana positiva por cada 30 ventanas negativas).

El otro parámetro que puede modificar el rendimiento es el número de ventanas muestreadas sobre la imagen. A continuación, evaluaremos el número de ventanas que se necesitan muestrear sobre una imagen en promedio para conseguir localizar el mayor número de objetos posible.

Las medidas de evaluación que se utilizan para determinar las prestaciones de los clasificadores propuestos son las siguientes:

- ***Precision***: Es el número de ventanas en las que se ha localizado correctamente un objeto (VP), dividido entre el número total de ventanas en las que se ha determinado la existencia de objetos, independientemente de si los había o no (VP+FP).

$$Precision = \frac{VP}{VP+FP} \qquad (4.1)$$

- ***Recall***: Es la fracción de objetos o instancias relevantes que se han localizado correctamente (VP) entre el número total de objetos que buscábamos localizar, tanto los que se han detectado como los que no (VP+FN). Dado un sistema de localización de objetos, posiblemente sea la medida más significativa de las tres que consideramos, pues informa acerca del porcentaje de objetos detectados.

$$Recall = \frac{VP}{VP+FN} \qquad (4.2)$$

- ***Accuracy***: Es el porcentaje de ventanas correctamente clasificadas por nuestro sistema, tanto aquellas que contienen objeto como las que se corresponden con regiones del fondo de las imágenes (VP+VN).

$$Accuracy = \frac{VP+VN}{VP+VN+FN+FP} \qquad (4.3)$$

4.4 Resultados y evaluación

A continuación, se presentan los distintos clasificadores, y sus correspondientes resultados para las medidas de evaluación descritas en la sección anterior.

Para los dos primeros sistemas SVM con kernel lineal determinaremos, en primer lugar, el valor óptimo de su parámetro C asociado, mediante el procedimiento de validación cruzada explicado en la sección 3.5.2. Para ello, realizaremos un barrido del mismo en escala logarítmica, de 2^{-3} a 2^{3}. Además, para las imágenes de los conjuntos de validación se considerará como referencia 2000 ventanas muestreadas. El tercer clasificador corresponderá a la fusión de los dos mejores clasificadores primero y segundo, es decir, aquellos con un valor de C óptimo escogido anteriormente.

A continuación, entrenaremos los sistemas atendiendo a ese valor del parámetro y, en la fase de localización o test del sistema, evaluaremos las prestaciones de las diferentes propuestas para el valor de C óptimo y un número determinado de ventanas muestreadas sobre las imágenes en las que se quiere localizar objetos.

Finalmente, para cada uno de los sistemas, se presenta un breve análisis de errores, con vistas a proponer líneas futuras de trabajo.

4.4.1 Resultados y evaluación del clasificador 1

En primer lugar, se determinará el valor de C óptimo para el primer clasificador, que toma como entrada las características DB, CC y ST, aplicando una validación cruzada con 5 particiones.

Para cada valor de C se tienen 5 resultados de validación, correspondientes a las cinco particiones establecidas. De esta forma se logra un resultado más preciso, al realizar la media de las medidas de evaluación obtenidas para cada una de las particiones.

A continuación, se muestra en una gráfica del recall obtenido en función del valor del parámetro C de la SVM validado, se ha escogido únicamente este parámetro ya que es el más representativo en la localización de objetos:

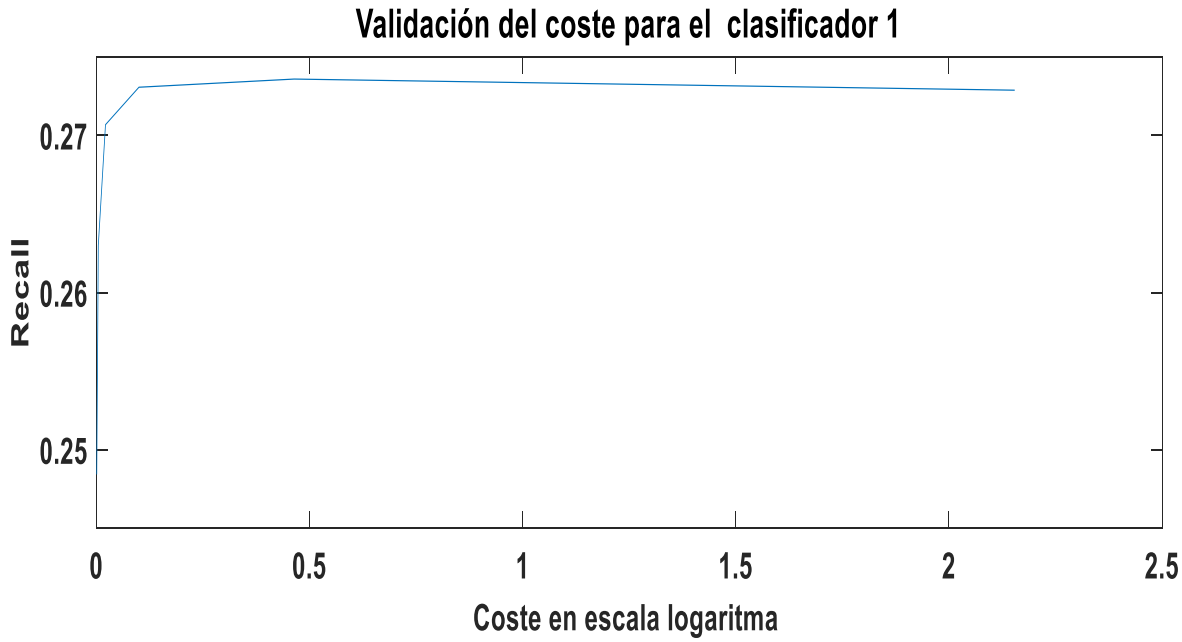


Figura 25. Validación del coste para el clasificador 1. El coste seleccionado es aquel que corresponda con un mayor recall

El valor de C óptimo, utilizado para obtener los resultados que se recogen a continuación, entrenando el clasificador sobre las ventanas del conjunto de entrenamiento de 1000 imágenes completo (40 ventanas por imagen), es 2^{-1}.

Una vez se tiene el modelo entrenado en su configuración óptima, se realizará la fase de test, donde el clasificador tratará de localizar objetos sobre las ventanas muestreadas a partir del histograma de localizaciones y relaciones de aspecto, en el conjunto de 2000 imágenes de test.

Para evaluar las prestaciones del clasificador, se irá aumentando progresivamente el número de ventanas muestreadas sobre las imágenes, de 100 a 2000 ventanas. De esta manera, tal y como se muestra en la Figura 25, podremos determinar un número de ventanas óptimo para localizar el mayor número de objetos posible, de acuerdo con las medidas accuracy, precision y recall definidas anteriormente.

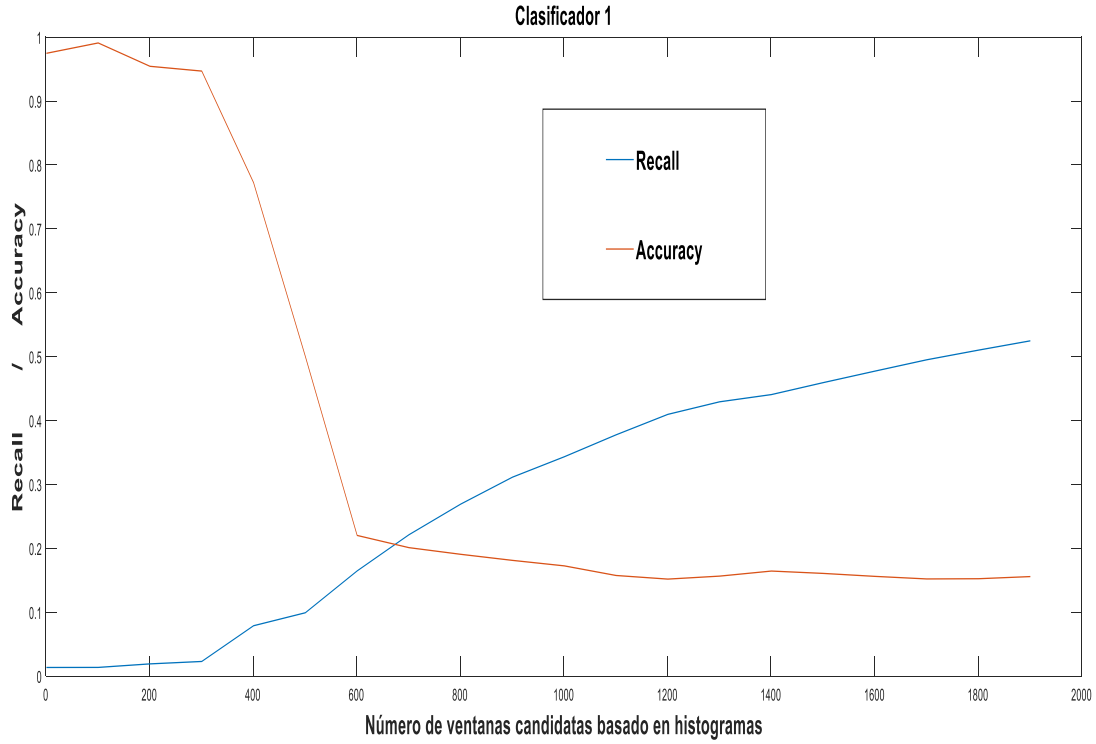


Figura 26. Gráfica Clasificador 1

Si nos fijamos en la Figura 26, se puede observar como el valor del recall óptimo es el que se da para un muestreo en torno a 300 ventanas y ese valor es equivalente a 0.96.

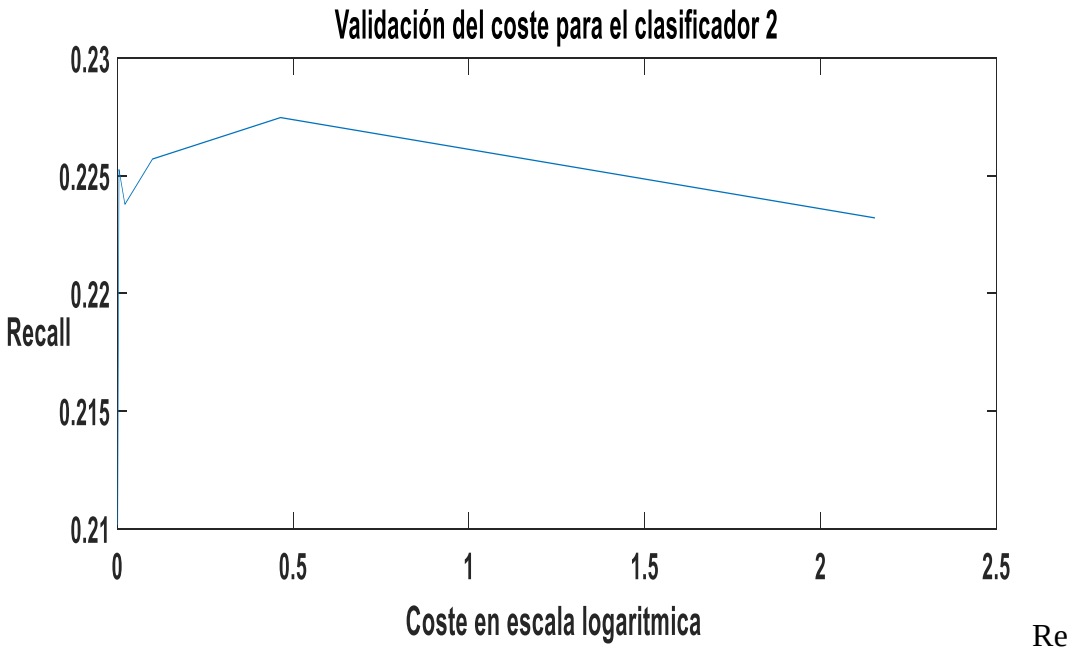
4.4.2 Resultados y evaluación del Clasificador 2

Los pasos a seguir en este segundo clasificador son los mismos que se han dado en el primero.

En primer lugar, se determinará el valor de C óptimo para el primer clasificador, que toma como entrada la característica NG aplicando una validación cruzada con 5 particiones.

Para cada valor de C se tienen 5 resultados de validación, correspondientes a las cinco particiones establecidas. De esta forma se logra un resultado más preciso, al realizar la media de las medidas de evaluación obtenidas para cada una de las particiones.

A continuación, se muestra en una gráfica la medida de evaluación recall obtenida en función del valor del parámetro C de la SVM validado. Se utiliza este parámetro únicamente por su mayor importancia en la localización de objetos:



Re

Figura 27. Validación del coste para el clasificador 2. El coste seleccionado es aquel que corresponda con un mayor recall

El valor de C óptimo, utilizado para obtener los resultados que se recogen a continuación, entrenando el clasificador sobre las ventanas del conjunto de entrenamiento de 1000 imágenes completo (40 ventanas por imagen), es 2^{-1}, coincidiendo con el clasificador 1.

Una vez se tiene el modelo entrenado en su configuración óptima, se realizará la fase de test, donde el clasificador tratará de localizar objetos sobre las ventanas muestreadas a partir del histograma de localizaciones y relaciones de aspecto, en el conjunto de 1000 imágenes de test.

Para evaluar las prestaciones de este segundo clasificador, se repetirá el mismo proceso que en el caso anterior. Se irá aumentando progresivamente el número de ventanas muestreadas sobre las imágenes, de 100 a 1000 ventanas. De esta manera, tal y como se muestra en la Figura 27, podremos determinar un número de ventanas óptimo para localizar el mayor número de objetos posible, de acuerdo con las medidas accuracy, precision y recall definidas anteriormente.

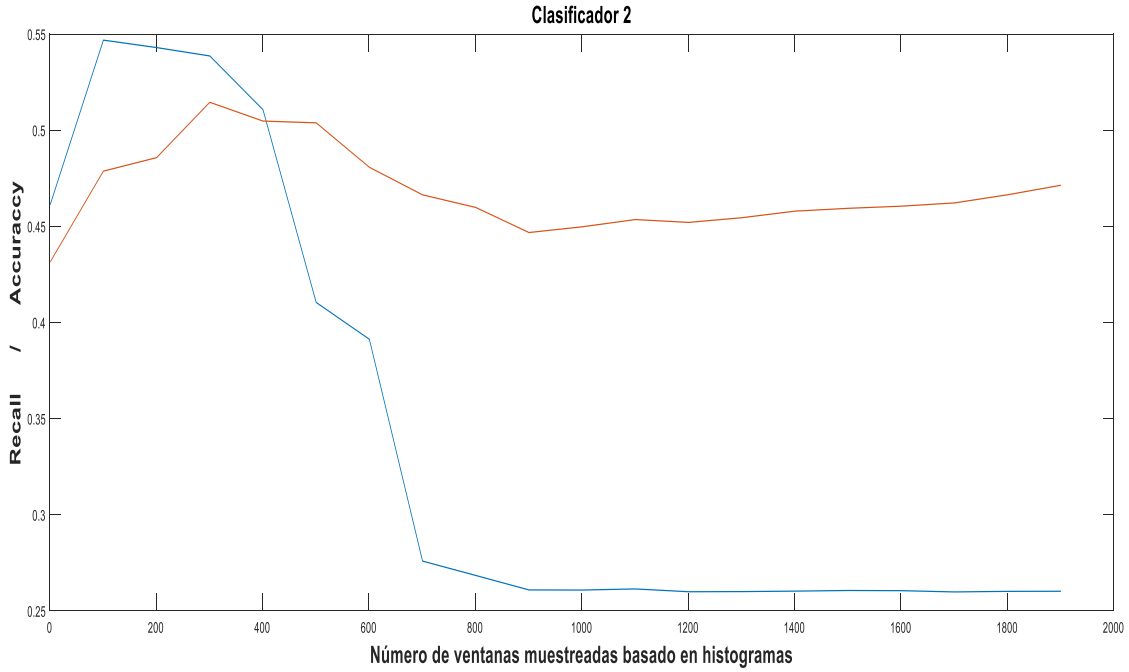


Figura 28. Gráfica clasificador 2

Observando la Figura 28, se aprecia que los resultados presentan un valor algo bajo en comparación con el clasificador ya visto, en torno al 50% de recall. Al igual que ocurría con los resultados para el clasificador 1, Figura 27, ésta muestra un mejor valor para un muestreo más bajo, y al aumentar el número de ventanas este resultado decae debido al aumento de falsos negativos.

4.4.3 Resultados y evaluación del conjunto completo

En este apartado se realizará la fusión de ambos clasificadores, para este punto es importante dejar claro que se han usado los modelos óptimos para cada sistema, obtenidas en los puntos anteriores.

Este clasificador consiste en utilizar los scores predichos tanto por el clasificador 1 como por el 2, y calcular la media de ambos. Una vez se tiene el nuevo score, se obtendrán las etiquetas de clasificación referentes a este sistema atendiendo al procedimiento que se ha explicado en la sección 3.5.3. De acuerdo con esta convención, una ventana con un score predicho mayor que 0.5 contendrá un objeto (clase 1, positiva), y una ventana con un score menor que 0.5 pertenecerá al fondo de la imagen (clase 0, negativa).

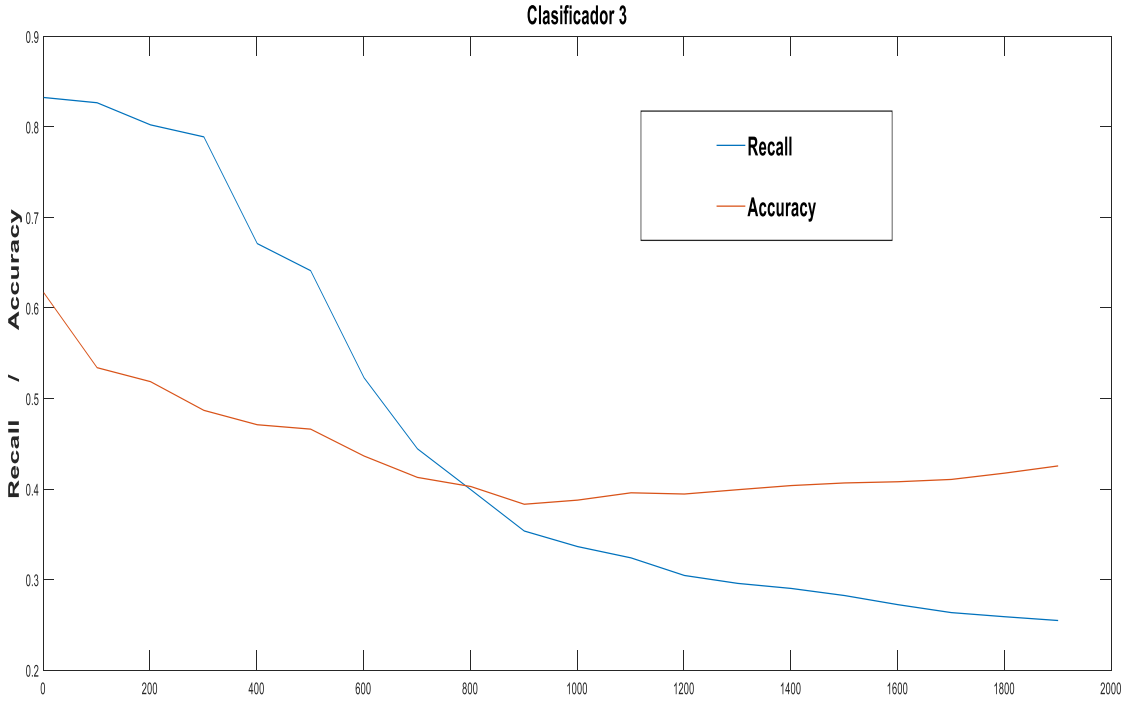


Figura 29. Grafica clasificador 3

Si se evalúa el sistema con el objetivo de que la unión de los scores, del clasificador de 3 features y el clasificador de una feature, dé un mejor resultado que ambos por separado, se puede concluir que la fusión no en beneficiosa ya que, si se comparan las Figura 27, 28 y 29, ya que el clasificador 1 ofrece unas prestaciones mucho más altas que el clasificador 2, por lo que el clasificador 3 no es capaz de mejorar el resultado del primer clasificador, con una fusión tan sencilla como la media de scores.

4.5 Análisis de los modelos

En este punto se valoran los pros y contras de todas etapas del proceso de detección y localización de objetos, para cada uno de los dos clasificadores configurados, no se tendrá en cuenta la fusión de ambos, ya que con el estudio de las partes por independiente es suficiente.

Se comenzará estudiando el clasificador 1, compuesto por las 3 features: densidad de bordes, contraste de color y superpíxels transzonales. Si se observa la Figura 28, se puede analizar que el comportamiento del clasificador es excelente, para los muestreos comprendidos entre 100 y 300 ventanas candidatas.

Esto nos lleva a catalogar como éxito el muestreo de ventanas candidatas basado en histogramas, ya que con pocas muestras el sistema es capaz de localizar objetos. Entonces se puede suponer que la bajada en el rendimiento del sistema se debe a un sobremuestreo del clasificador, porque el número de ventanas es mayor al debido.

Para estar más seguros, estudiaremos algunos casos dados en el clasificador, donde el valor del recall es más bajo. En la Figura 30, se puede ver un par de ventanas candidatas que han sido clasificadas como positivas, y efectivamente contienen un objeto, en este caso una moto. Si se presta especial atención al score, se observa que son valores muy próximos a 0,5, lo que supone que una ventana similar, pero con una pequeña variación de localización o tamaño será catalogada como fondo, siendo por lo tanto una FN, ya que si hay objeto.

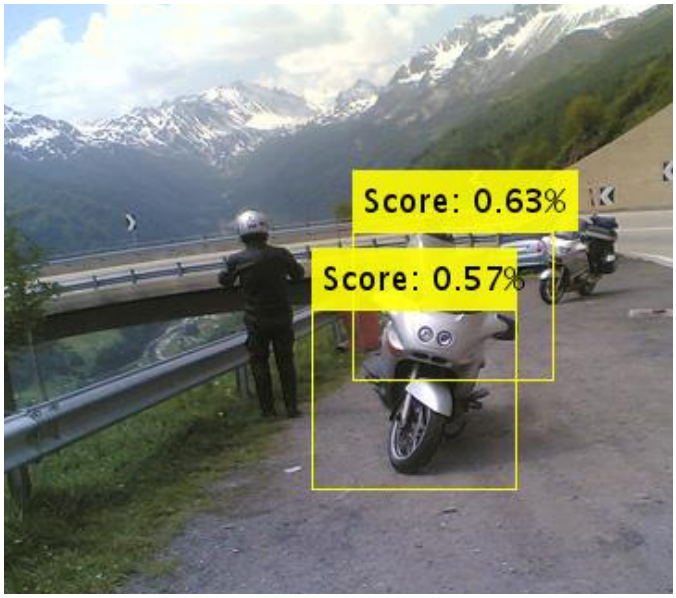


Figura 30. Ejemplo VP

Como comprobación seleccionamos la misma imagen, y buscamos una ventana que haya sido catalogada como negativa, fondo. Pero que, si contenga objeto, un FN, esa ventana se puede ver en la Figura 31. Si se hace una pequeña comparación, tan solo mirando ambas imágenes, es difícil decir porque no tienen la misma clasificación sí parece que ambas contienen el objeto, la moto.

Figura 31. Ejemplo FN

Si se realiza una comparación más precisa, se concluye que mientras que la ventana catalogada como positiva tiene un valor de score de 0.57, la que ha sido catalogada como negativa tiene uno de 0.48. Por lo tanto, ahora sí se puede afirmar que el clasificador 1, detecta con un gran nivel de recall los objetos, aunque para ello las ventanas candidatas basadas en histogramas deben estar dentro de un valor óptimo.

Por el contrario, el bajo valor del recall en el clasificador 1, podemos concluir que se debe a un sobremuestreo, ya que con pocas muestras las ventanas candidatas basadas en histograma ya encuentran al objeto. Y la generación en exceso de ventanas hace que haya un valor de score más próximo a 0.5

Una vez se ha analizado el clasificador 1, se procede a realizar el mismo estudio con el clasificador 2. Si se analiza el valor del recall en función de las ventanas muestreadas, como se recoge en la Figura 29, se observa el mismo patrón que el clasificador 1. Cuando el muestreo de ventanas candidatas generadas por histograma, oscila entre 100 y 400 ventanas, se tiene un valor bastante diferente, a cuando es entre 800 y 2000.

No podemos concluir que la causa de la caída del valor del clasificador 2 sea la misma, que la del clasificador1. Para encontrar el problema se realizará el mismo proceso que con el clasificador 1, ya que presenta las mismas condiciones. En primer lugar, se selecciona una muestra que haya sido catalogada correctamente como objeto.

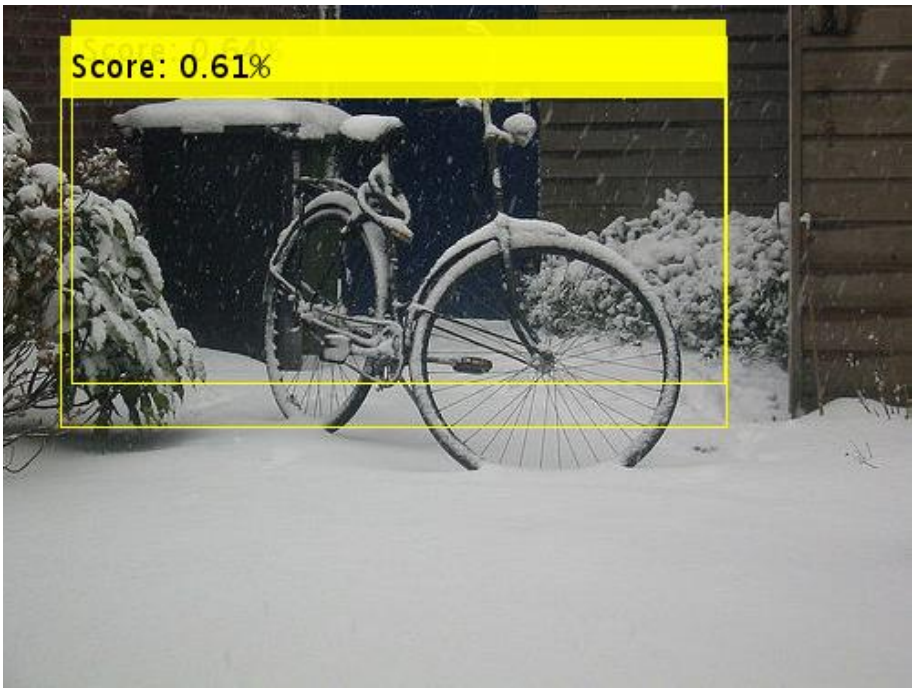


Figura 32. Ejemplo VP clasificador 2

Si se estudia el caso seleccionado como VP, correspondiente al clasificador 2, se observa que la ventana candidata localiza correctamente el objeto. Respecto al valor del score, se puede indicar que es algo próximo a 0.5, la frontera de decisión del clasificador. Para ver el comportamiento de una ventana que contenga un objeto y haya sido clasificada como fondo, visualizaremos un FN.

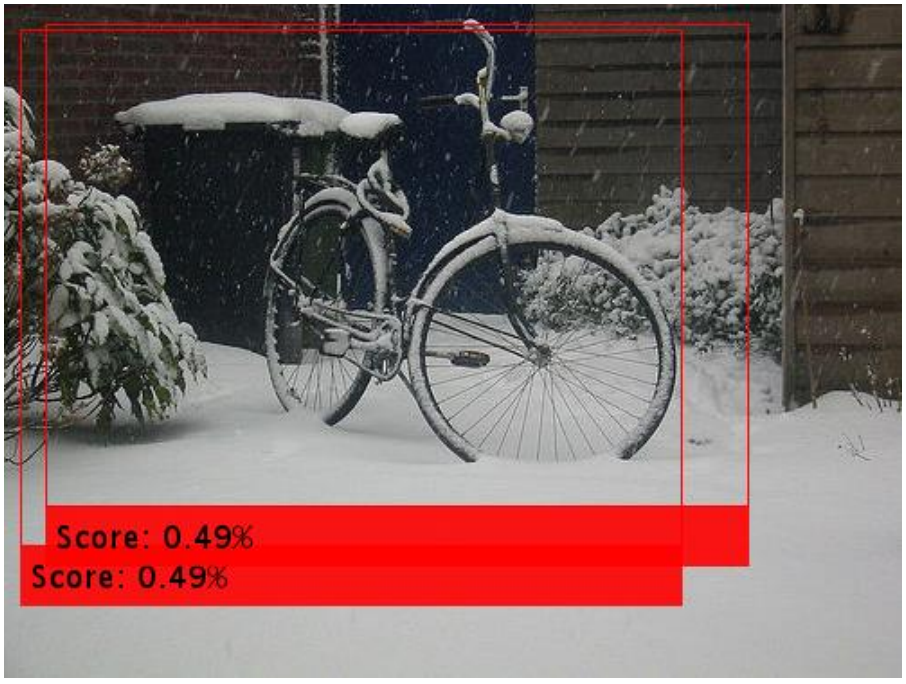


Figura 33. Ejemplo FN clasificador 2

Visualizamos la Figura 33, para aplicar nuestro análisis sobre un par de muestras mal clasificadas. Si nos basamos en el contenido de la ventana, se puede decir que el objeto se encuentra

correctamente localizado en su mayoría, no se trata únicamente de una parte del objeto, o de una ventana de dimensiones muy distintas al objeto. Si analizamos el valor de los respectivos scores para cada una de las ventanas, se observa que se está ante un caso límite. El valor de ambos scores es el mismo, 0.49, siendo la frontera de decisión 0.5, se puede concluir que el excesivo muestreo de ventanas candidatas basado en histograma, disminuye el valor de los scores lo que dificulta su clasificación.

Es lógico que, si ambos sistemas presentan una considerable disminución de su parámetro recall, en un número similar de ventanas de muestreo, el problema para ambos casos sea el mismo. Como ya se explicó al principio de la sección, no se evaluará el clasificador 3, ya que se trata de una fusión de los que ya se han evaluado, y su funcionamiento de representa ninguna mejora de rendimiento respecto del clasificador 1 por separado.

5. PLANIFICACIÓN DEL TRABAJO Y PRESUPUESTO

5.1 Introducción

En este capítulo se muestra, en primer lugar, un esquema de la organización seguida para la realización del proyecto. A continuación, en la sección siguiente, un gráfico de Gantt, para visualizar el tiempo de trabajo previsto y asignado a las diferentes actividades a lo largo del tiempo estimado de realización del proyecto.

5.2 Planificación del trabajo

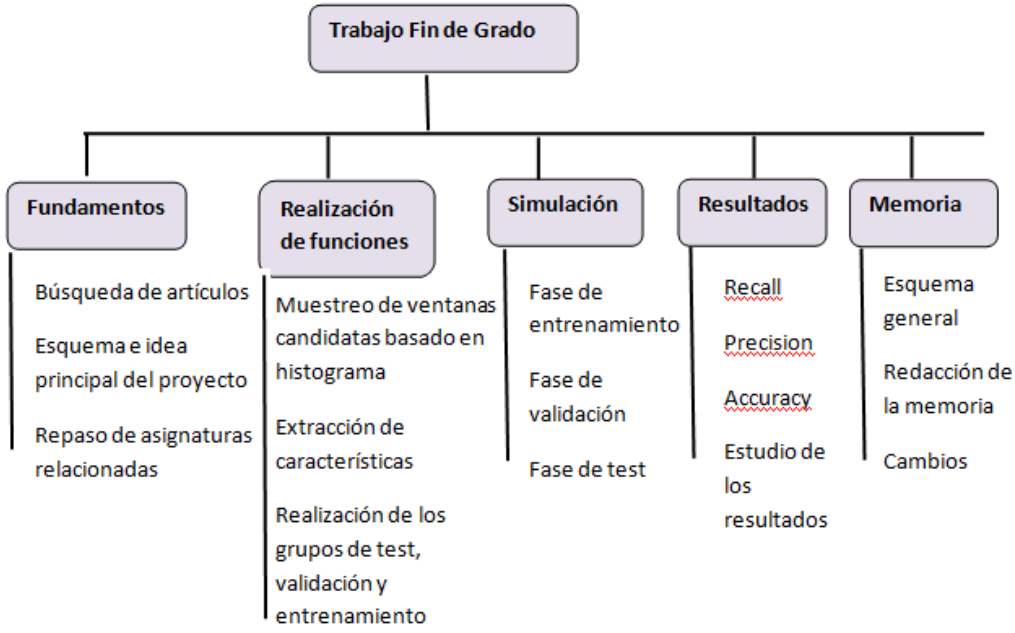


Figura 34. Planificación del trabajo

5.2.1 Diagrama de Gantt

Tarea 1. Búsqueda de información y artículos relacionados

Tarea 2. Organización y definición de las etapas del proyecto

Tarea 3. Muestreo de ventanas a partir de histograma

Tarea 4. Extracción de características

Tarea 5. Fase de Entrenamiento y validación del clasificador

Tarea 6. Fase de Test

Tarea 7. Evaluación de resultados

Tarea 8. Realización de la memoria

Tarea 9. Cambios y optimización de resultados

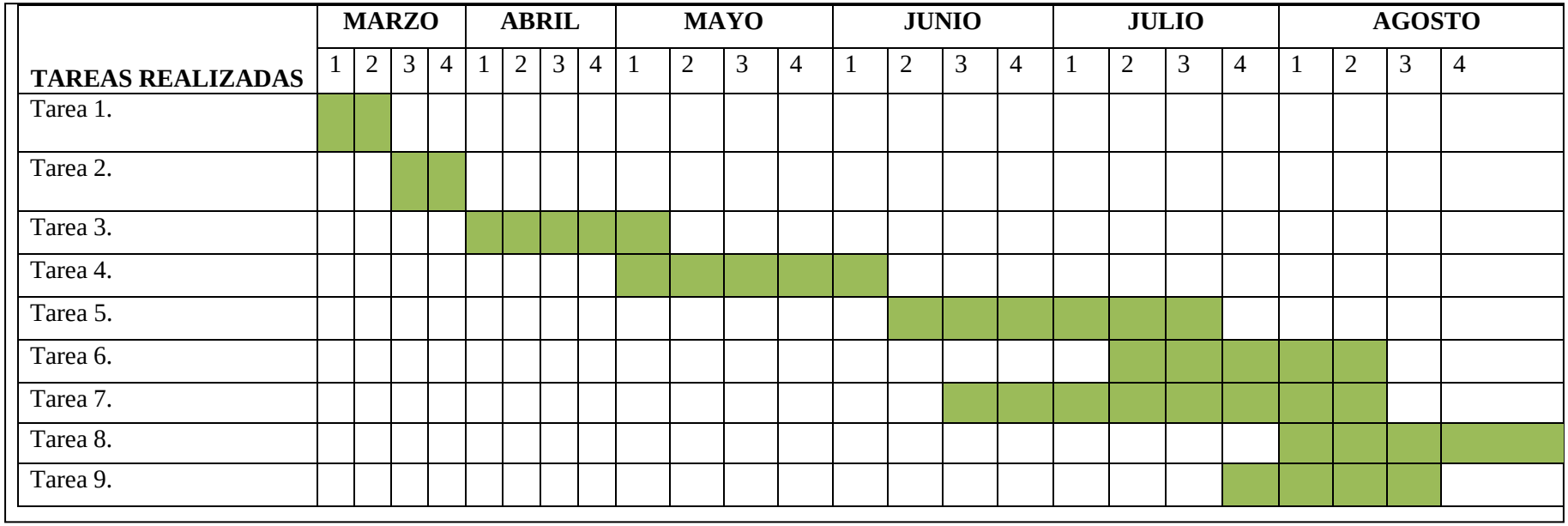


Figura 35. Diagrama de Gantt

5.3 Presupuesto

Para el planteamiento del presupuesto correspondiente a la realización del sistema presentado, es importante tener en cuenta que este proyecto consiste en la ejecución y simulación computacional en el ámbito de la localización de objetos, por lo que los principales activos son los equipos encargados de estas tareas, los softwares empleados y las personas encargadas de llevarlo todo a cabo. Por este motivo, algunos campos habituales en los presupuestos no serán tenidos en cuenta.

Costes Fijos

Son los costes que permanecen invariables ante el cambio en los niveles de producción.

- Devaluación del valor del ordenador por su utilización en el uso de ejecuciones y demás tareas relacionadas con el trabajo: Se establece aproximadamente que la vida útil del equipo es de unos 10 años. Con un precio original de 600€, su tasa de depreciación es de 60€/año. Como se ha establecido en el punto 5.2 la duración del proyecto se establece en torno a 6 meses, por lo que la devaluación sufrida durante este periodo de tiempo debida al uso por el trabajo es de 30€.

-Licencias de software Durante el desarrollo del proyecto se ha empleado el programa MATLAB cuya licencia de pago depende de los paquetes que solicitemos, el paquete básico son 20€ pero no consta de la librería de funciones para Machine Learning, algo que es necesario para el proyecto. Por si algún otro paquete fuera necesario, se compra la versión completa que, cuyo precio se estima en 69€. Esto no ha sido necesario gracias al acuerdo de la universidad con la empresa, pero para el cálculo del presupuesto no se considera este acuerdo. La base de imágenes es una base gratuita PASCAL VOC [11], formada por estudiantes y otras personas que investigan campos relacionados con las imágenes y se ayudan entre sí, así que a efectos de presupuesto no supone ningún gasto.

Costes Variables

Son costes que irán cambiando su valor en función del trabajo realizado.

- Energía eléctrica consumida, depende de las horas de trabajo asociadas al proyecto. Para el cálculo de la energía se va a establecer que se han empleado únicamente dos elementos que necesiten de la misma. en este caso son el ordenador y una lámpara, que consumen una potencia de 120W y 50W respectivamente. Para establecer el tiempo que se han empleado ambos dispositivos se considera que el ordenador ha estado activo toda la jornada de trabajo mientras que la lámpara solo la mitad del horario. El número total de horas de trabajo es aproximadamente 400 horas.

Para establecer la energía consumida, se consulta la página oficial Iberdrola [29] y se fija como precio actual de la luz 0.134 €/kWh se establece que se ha mantenido constante

durante la realización del proyecto, por lo que el precio final de la factura de la luz es de 7,77€ de los cuales 6,43€ consumidos por el ordenador y 1,34€ consumidos por la lámpara.

Para estimar la cantidad económica retribuida al alumno y al tutor se utilizan valores estimados de los sueldos de ambos. Para el alumno se fija un sueldo medio de prácticas de 7€ la hora, y se ha estimado que las horas dedicadas al trabajo han sido de 400, por lo que la retribución del alumno es de 2.800€. El sueldo medio para el tutor se establece en 18€ la hora, se estima que las horas invertidas por parte del tutor en la elaboración del trabajo es de unas 50, por lo que su retribución es de 900€.

Tabla 2. Tabla de presupuestos del trabajo

Coste €	
Depreciación del equipo	60 €
Licencia MATLAB	69 €
Licencia PASCAL VOC	Gratuito
Energía eléctrica	7,77 €
Retribución tutor	900 €
Retribución alumno	2800 €
Total	**3.836,77 €**

6. CONCLUSIONES Y LÍNEAS FUTURAS

6.1 Conclusiones

El propósito de este proyecto ha sido el de estudiar la efectividad de la localización de objetos mediante la extracción de características sobre ventanas muestreadas en una imagen digital.

El primer factor que se ha tenido en cuenta ha sido el de definir una técnica eficiente de muestreo de ventanas candidatas, capaz de posicionarlas en las zonas donde más probabilidades hay de localizar un objeto. Tras el estudio de procesos ya existentes, se decidió implementar un método alternativo; un muestreo de ventanas candidatas basado en un histograma de localizaciones, determinadas por la regla de los tercios, y relaciones de aspectos de un conjunto de objetos conocidos a priori.

Este método, efectivamente, consigue generar ventanas en aquellos sectores donde es más probable que se encuentre el objeto. Por ejemplo, es más probable que un objeto se encuentre en un sector centrado, que en el sector de la esquina inferior derecha.

Aun así, el muestreo también genera ventanas en zonas de menor probabilidad, en menor medida, de manera que, en caso de que una imagen contenga un objeto en una zona menos probable, exista todavía cierta posibilidad de encontrarlo.

En resumen, se realiza un muestreo completo de la imagen, dando una mayor importancia a las zonas donde es más posible que se encentre un objeto, en este punto se ha tenido bastante éxito. Por el contrario, la dimensión de las ventanas generadas ha supuesto algunos inconvenientes para la localización de objetos debido a que, si la ventana no acota adecuadamente la zona del objeto, descriptores como la norma de gradiente NG resultan ser menos efectivos de lo esperado.

En segundo lugar, la fase de extracción de características ha tenido como referencias técnicas usadas en otros estudios [6,7,12], basadas en aspectos característicos del objeto con respecto al fondo de la imagen, como el contraste de su color, su contorno continuo y en muchas ocasiones compacto, y la densidad de bordes alrededor del mismo.

A la vista de los resultados ofrecidos por las alternativas de diseño propuestas, se puede concluir que los descriptores de contraste de color y superpíxeles transzonales ofrecen unos resultados muy buenos cuando se utilizan a la entrada de una SVM con kernel lineal. Además, el método de muestreo de ventanas basado en el histograma propuesto permite localizar rápidamente las zonas más probables a objeto de las imágenes, reduciendo considerablemente el número de ventanas utilizadas para detectar los objetos.

6.2 Líneas futuras

En este apartado se tratarán las posibles modificaciones que podrían sufrir los bloques del sistema en un futuro, pero siempre intentando lograr el mismo objetivo.

Para enumerar las líneas futuras se tendrán en cuenta, una vez más, las diferentes etapas del sistema.

Comenzando por el proceso de muestreo, es este proyecto se ha fundamentado en histogramas. Pero no es la única forma posible de muestrear como se mencionó en el capítulo 3, sección 3, también se podría añadir más información sobre la fotografía para capturar los objetos muestreando un menor número de ventanas. Esta información adicional podría sacarse de los colores de la imagen o de descriptores basados en su textura, como se realiza en [28].

Con respecto a la siguiente etapa del sistema, la extracción de características, después de un estudio del tema y un entendiendo suficiente de la materia se podrían llegar a usar redes neuronales convolucionales (Convolutional Neural Networks, CNNs). Tratando de imitar el comportamiento de las neuronas en el cerebro de los seres humanos, estas redes permiten extraer características de muy alto nivel que facilitan la representación de zonas muestreadas en la imagen, mejorando notablemente los resultados que ofrece un sistema de localización de objetos más tradicional, como el que se presenta en este trabajo.

Existen otras posibilidades de mejorar los resultados obtenidos sin tener que cambiar el proceso de muestreo ni la extracción de características. Estos cambios se producirían en las etapas de clasificación del sistema.

Una posible medida es seleccionar aquellas muestras más difíciles de clasificar por el sistema, dada una etapa de validación. Con estas muestras más difíciles de clasificar se volvería a entrenar el sistema, esto se denomina re-entrenamiento [22], adaptando el hiperplano de clasificación inicial a estos nuevos datos.

También se podría cambiar la configuración de la SVM atendiendo al kernel utilizado [23]. En este proyecto, por simplicidad, sólo se han probado kernels lineales, pero otros, como el frecuentemente usado RBF, o incluso un kernel adaptado a características de tipo histograma, como la norma de gradiente que usamos en la segunda arquitectura propuesta, podrían ser de utilidad para mejorar la eficiencia del sistema.

Por último, pero no menos importante, los experimentos presentados en este proyecto se han realizado utilizando un conjunto de imágenes más pequeño que los habituales en bases de datos para problemas de localización de objetos. Por tanto, sería interesante realizar un análisis más exhaustivo de las alternativas presentadas, considerando un conjunto mayor de imágenes.

7 BIBLIOGRAFÍA

[1] Bradski, Gary R. "Computer vision face tracking for use in a perceptual user interface." (1998).

[2] Martínez Arcila, Héctor Fabio, Gaitán Tabares, and David Leonardo. "Sistema Autónomo para Recolección de Bolas de Tenis Mediante Visión Artificial." (2016).

[3] Tornero Montserrat, Josep, et al. "Detección de defectos en carrocerías de vehículos basado en visión artificial: Diseño e implantación." *Revista Iberoamericana de Automática e Informática Industrial RIAI*. Vol. 9. No. 1. Universitat Politecnica de Valencia, 2012.

[4] Cuenca, Julián Sanz. "Reconocimiento de objetos por descriptores de forma." *Departamento de Mátematicas Aplicada y Análisis, Universidad de Barselona* (2008).

[5] Viola, Paul, Michael J. Jones, and Daniel Snow. "Detecting pedestrians using patterns of motion and appearance." *International Journal of Computer Vision* 63.2 (2005): 153-161.

[6] Alexe, Bogdan, Thomas Deselaers, and Vittorio Ferrari. "What is an object? " *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.

[7] Farhadi, Ali, et al. "Describing objects by their attributes." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

[8] Siva, P., Russell, C., Xiang, T., & Agapito, L. (2013). Looking beyond the image: Unsupervised learning for object saliency and detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3238-3245).

[9] Alexe, Bogdan, Thomas Deselaers, and Vittorio Ferrari. "Measuring the objectness of image windows." *IEEE transactions on pattern analysis and machine intelligence* 34.11 (2012): 2189-2202.

[10] Enlace de la regla de los tercios: https://www.dzoom.org.es/regla-de-los-tercios/ . Consultado el 18 de septiembre de 2018.

[11] Joshi, N. A., and J. N. Fass. "Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ files (Version 1.33)[Software]." (2011).

[12] Cheng, Ming-Ming, et al. "BING: Binarized normed gradients for objectness estimation at 300fps." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[13] Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." *Data mining and knowledge discovery*2.2 (1998): 121-167.

[14] System Requirements for MATLAB & Simulink R2016b (Professional & Student Versions). MathWorks. Consultado el 20 de septiembre de 2018.

[15] Geronimo, David, et al. "Survey of pedestrian detection for advanced driver assistance systems." *IEEE transactions on pattern analysis and machine intelligence* 32.7 (2010): 1239-1258.

[16] Lu, Wenjun, Avinash L. Varna, and Min Wu. "Security analysis for privacy preserving search of multimedia." *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010.

[17] Farfade, Sachin Sudhakar, Mohammad J. Saberian, and Li-Jia Li. "Multi-view face detection using deep convolutional neural networks." *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015.

[18] Urmson, Chris, et al. "Autonomous driving in urban environments: Boss and the urban challenge." *Journal of Field Robotics* 25.8 (2008): 425-466.

[19] Enlace función cálculo de distancia chi-cuadrada. https://es.mathworks.com/matlabcentral/fileexchange/8891-randp-p-varargin

[20] Fernández Sánchez, Manuel Carlos. "Hacia una teoría complementaria del encuadre." *Comunicación. Revista Internacional de Comunicación Audiovisual, Publicidad y Literatura* 1.1 (2002): 89-98.

[21] The PASCAL Visual Object Classes Challenge: A Retrospective Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A.*International Journal of Computer Vision, 111(1), 98-136, 2015*

[22] Lum, Peter S., Charles G. Burgar, and Peggy C. Shor. "Evidence for improved muscle activation patterns after retraining of reaching movements with the MIME robotic system in subjects with post-stroke hemiparesis." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 12.2 (2004): 186-194.

[23]es.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html#bsr5b42 . Consultado el 18 de septiembre de 2018.

[24] Müller, Timo D., et al. "Restoration of leptin responsiveness in diet- induced obese mice using an optimized leptin analog in combination with exendin- 4 or FGF21." *Journal of Peptide Science* 18.6 (2012): 383-393.

[25] Harrell, Frank E. "Ordinal logistic regression." *Regression modeling strategies*. Springer, Cham, 2015. 311-325.

[26] Deming, W. Edwards, and Frederick F. Stephan. "On a least squares adjustment of a sampled frequency table when the expected marginal totals are known." *The Annals of Mathematical Statistics* 11.4 (1940): 427-444.

[27] Enlace Figura 7. http://adolfoalfonzo.blogspot.com/2015/11/comunicacion-del-color.html. Consultado el 18 de septiembre de 2018.

[28] Uijlings, Jasper RR, et al. "Selective search for object recognition." *International journal of computer vision* 104.2 (2013): 154-171.

[29] Enlace página official de Iberdrola, https://www.iberdrola.es/. Consultado el 18 de septiembre de 2018.

7.1 Legislación y jurisprudencia

[A] BOE-A-1982-11196

ANEXO 1. SUMMARY

In the last years, Computer Vision area has become one of the most important fields of research. In addition, these methods can be applied in many other fields because of advances in the technology.

This development has allowed the incorporation of elements that previously could be high cost or difficult to make, like the automotive sector, mobile telephony and home security.

We can define Computer Vision as a part of Artificial Intelligence, that studies the development of techniques for the processing and analysis of the features extracted from the digital images. The most important applications of Computer Vision are:

- Identification and analysis of objects.
- Location of objects in the scene.
- Shaping and reconstruction of 3D objects.
- Screening and redevelopment of specific parts from an object (corners, edges, etc.)

The project that it is going to be presented consists of the study and development of a system for the location of objects in images, that will sample some windows on each image, extracting characteristics that allow their subsequent classification, depending on whether having an object or if it belongs to the background of the scene.

There are many sectors depending on the objective to be investigated, from large dimensions like a monument to something as small as a logo in a jacket. An example of a real application could be a digital camera where it is looking a facial recognition for the correct identification of a person. Another simpler case is the screening of the barcode of a product, always identifying the rectangle around the barcode.

The most popular use could be the technology of 'hawk eye' used in the tennis. It is based on locating the tennis ball when bouncing on the track, that is why we can know if the complete ball is outside of the limits of the track, or either conversely it has not left completely. This system is based on the live taking of some pictures and next triangulation of the final position of the object, the tennis ball in this case. In the same way this system is trying to be adapted to other sports, like football, because of its high level of accuracy.

One of the areas with most researches is driving safety, for example, the 'environment detection system', a system that is being more and employed in cars to prevent accidents or avoid people or cyclists being run over by the driver for any cause. For this system it is necessary to use at two high resolution cameras placed somewhere to watch all around.

The main objective of this project is the detection and location of objects in images. For that aim we will study different descriptors and algorithms, with the idea of comparing

different options and find those technologies that produce better benefits. The project can be described according to the following specific goals:

1. Sampling: For the correct extraction of features, to begin we must get the right sampling. It consists in the selection of windows the most representative possible of the all image. If it is possible we will do it with few windows.

2. Achieve data based exclusively on the information about the image: In order to demonstrate that it is possible to do so by using the appropriate features to get a high level of precision to locate the object as result, as long as we only use the features of photography and its information.

3. Feature extraction: The best possible case in that feature to help us differentiate between object and background, for example, a cat as object, and sky as background. Moreover, features are going to be used to teach the classifier, to decide if is there is some object or not, however the more the system learns the better the results will be.

4. Evaluation: We must study the results obtained by the classifier. This is the reason why we must use so many images, so that it is representative. The PASCAL VOC database has been used for this.

The problem

The problem of the detection and localization of object has been studied during the last decades, but in all studies we can find some common things. In order to locate an object from a known class, the most developed technique is based on the characteristics of the element.

We are going to explain with a simple example what the characteristics of an object consist of. Firstly, we choose some characteristic (color, size, shape, etc.) and we describe some objects according to them.

- Basketball ball: its color is orange, round shape and size is 65-70 cm.
- Television: its color is black, rectangle shape and size 32 inches.

Conversely if we want to identify a general object we don't use this characteristics, because if searching for an orange object, it won't find the television. In others words, if we want detect and find any type of object, we must take properties common to all, allowing to differentiate them from the background of the image regardless of its characteristics.

In this project we want to locate the ball and the television, so we cannot base the search on the object own characteristics (color, shape, size, etc.). We must look for common features for all objects, like both they have a defined body, yet not the same shape, yet it is a closed and defined form.

In general, the objects have a different color to the background, it is not important what it is different between them, the important idea is that the color of the object is different from the background. For example, if we have a basketball ball and a television on the grass, we will find them because they are not green.

Sampling of candidate windows based on histograms

The first challenge to face was to determine the position for the windows that were candidates to contain an object in the image. In most projects the solution consists on using *sliding windows*, that draw windows in every pixel of the images, and with different sizes, for being sure to find the object. In our project, however, it is not used this method but another more efficient solution, that learns the most frequent position for the objects and the most frequent size of the window that must contain the object, to just draw windows in those position and sizes.

From a big set of images with the position and size of the objects in them, it builds a histogram based on that location (Ground-Truth). The system divides the area of the image in 9 quadrants, using the rule of thirds. According to the objects' Bounding Boxes data (coordinates of the Bounding Boxes surrounding objects: upper left corner coordinates given by *x* and *y*, and size, given by *width* and *height*), the system labels each quadrant of the image as 1 (containing an object) or 0 (does not contain an object).

Then, for each image we can get a histogram of which quadrants do contain an object, so that for a big number of images, the joint histogram shows the quadrants in which it is most likely to find objects. We have implemented this sampling over 5,000 images from a sufficiently representative and generic set.

Feature extraction

For extracting the features and characteristics of the objects in the image we have tested different solutions:

Edge Density

This technique is based on detecting the object after its edges, assuming a well-defined structure. In the greyscale image, the process reduces the size of the

candidate window around the inner ring of the image. Then it calculates the edge density in that inner ring by using Canny's detector.

Color Contrast

Color Contrast feature extraction is based on differentiating objects according to their color, after color space LAB definition. The CC is the measure of dissimilarity between a window and its immediate environment.

The contrast difference is calculated by the Chi-Square distance between the histograms of the outer ring and the ones of the candidate window. For extracting those histograms, the system applies two masks (one for each case) to separate the two areas.

Superpixels Straddling

As in the Edge Density description, this system identifies possible objects by characterizing closed limits. The fundamental principle is that all the pixels in a superpixel belong in the same object, and that object is divided in a finite set of superpixels, so that the objective is to find those superpixels that grouped form the object.

According to this, if an object is surrounded by the candidate window, most of the superpixels will be fully inside of it, and if the window doesn't contain an object, most of superpixels are split by the window edge.

The Superpixels Straddling descriptor gives as result the degree to which all the superpixels that have some of its pixels inside the window and some outside, ignoring the superpixels fully contained or excluded from the window.

Normal Gradient

This descriptor differentiates the object from the background after its borders by the gradient of the image and its module.

Classification

The classification consists on labelling the elements of the sample set between N different classes with their own characteristics. Among the different classification systems there are supervised learning algorithms and unsupervised learning algorithms. In this project we are implementing a supervised learning algorithm.

The classifier takes three steps: training, validation and test, that must be executed in that order. We implemented the crossed-validation technique, that consists of alternating part of the samples as training and validation in a finite number of iterations (in five, in this project). In other words, the training data with its correct labels get split in five sets, and each time four of them are used to train the model and the other one to evaluate the quality of the result of the learning in its own set.

In the training phase, the system receives a set of characteristics obtained in the previous phase of feature extraction with their correct labelling, in order to learn from them some common points between all the objects characteristics. As crossed-validation is implemented, for this training phase it will receive 4/5 of the training examples each time, and it will alternate which of them are for this training and which are for the evaluation.

Then, with the properties the system learned as proper to objects, the system gets a new set of properties, and tries to classify them as proper of object or background. If we are yet in the crossed-validation, it will compare the results of this 1/5 of the training data with the correct labelling in order to get the features and know how good it was.

By last, the validation process gets executed again with the test set, so it labels the data as object or non-object, but this time as it is the test set, it doesn't have the true labels, so it cannot compare to the correct solution. This would be an example of its working in a real scenario, where the correct answer is unknown.

Results

In this project we evaluate the efficiency with which the classifiers that have been defined are able to assign the correct labelling to any set of data, regardless of the classes that make up the samples. For that aim we have used the crossed-validation data and the test-phase results.

We have calculated different measurements:

- False Positive (FP), case in which it finds an object that is not actually an object.
- True Positive (TP), true object detected.
- False Negative (FN), object missing and not being detected.
- True Negative (TN), no-object, correctly labelled as no-object.

With those parameters we can calculate the evaluation measurements:

- Precision, measures the relevant labels (true object labelled as so) among all those labelled as object-containing.

- Recall, measures the relevant labels (true object labelled as so) among all the true objects (labelled as object and as no-object).
- Accuracy, measures the preciseness with which the samples are classified among each class, so that all the true labelled samples are divided between all the labelled samples.

Conclusions

The first step in this project was defining a sampling technique in order to locate the objects in the picture in a more efficient way than the classic sliding windows method. For that aim our system predicts the most likely sizes and locations for the objects, so that it draws more windows in that areas with that size (though it also draws windows in other locations and sizes, to make sure all objects are found and not only the most predictable ones. This worked optimally for the location of the windows, but not so much for the sizes of the windows, that were more varied.

Future lines

Among some possible improvements for this project, we could study some other techniques to determine the position and size of the windows that were candidates to contain objects, besides the histogram-based method we implemented. Another system could extract some other information from the picture, for instance, the luminance of the image, or the colors of it.

In the feature extraction we could use convolutional neural networks, that are artificial neural networks in which every neuron corresponds to receptive fields very similar to those of a human brain.

There are other possibilities to improve the results obtained without having to change the sampling process or feature extraction. These changes would occur in the training and validation stages of the system. With the incursion of the modifications it is sought that the classifier improves its learning with the data that is already used in this project, without modifying anything in them.

A possible action is to select those samples that are more difficult to classify by the system in the validation stage, attending to the values which are closer to the classifier limits, and re-training with those samples.

Other variations that can be made very simply are varying the classifier parameters. Although several configurations have been used in this project and the best result has been chosen, not all fields have been explored. The classifier that has been used

throughout the project (Support Vector Machine, SVM) classifies using the Kernel setting, but there are other possible configurations.

Beyond the work stages of this system there are other parameters that could be changed. One of the simplest, but no least, is to increase the number of images in the database. With this, the system will have more samples to train with, and therefore more samples from which to learn. The use of a different database could also be a point of interest, to check if an increase in the classes or in the way of taking the pictures modifies the results by applying the same procedure.